BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problems Mailbox.

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平9－134406

(43)公開日　平成９年(1997)５月20日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｋ　9/20	３４０		Ｇ０６Ｋ　9/20	３４０Ｊ	
Ｇ０６Ｔ　11/60			Ｇ０６Ｆ　15/62	３２５Ｄ	

審査請求　未請求　請求項の数51　ＯＬ　（全 61 頁）

(21)出願番号　特願平7－341983

(22)出願日　平成７年(1995)12月28日

(31)優先権主張番号　特願平7－229508

(32)優先日　平７(1995)９月６日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　勝山　裕
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　直井　聡
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　大菅　義之　（外１名）

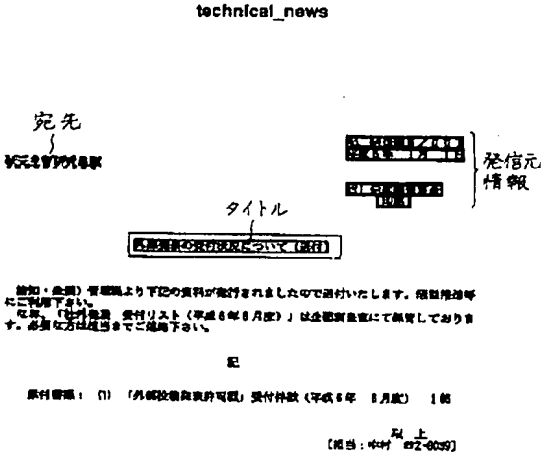
(54)【発明の名称】　文書画像からのタイトル抽出装置および方法

(57)【要約】

【課題】　文書を画像データに変換して得られる文書画像から容易にタイトル部分を抽出することが課題である。

【解決手段】　タイトル抽出装置は、文書画像内の黒画素を走査し、それらが連結している領域に外接する矩形領域を文字矩形として抽出し、さらに、隣接する複数の文字矩形を統合して、それらの文字矩形に外接する矩形領域を文字列矩形として抽出する。次に、各文字列矩形の下線属性、枠付き属性、罫線属性等の属性と、文書画像内の文字列矩形の位置や相互の位置関係とに基づいて、タイトルらしさのポイント計算を行い、高ポイントを獲得した文字列矩形をタイトル矩形として抽出する。また、表形式の文書の場合、表内からタイトル矩形を抽出することもできる。抽出されたタイトル矩形内の文字は、認識処理後に文書画像のキーワードとして用いられる。

タイトルおよび宛先・発信元情報の他の抽出結果を示す図



1

【特許請求の範囲】

【請求項１】　文書を画像データに変換して得られる文書画像から必要とする部分領域を取り出して認識する情報処理装置において、

前記文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成する文字領域生成手段と、

前記文字領域生成手段が生成した１つ以上の文字領域を統合して、該１つ以上の文字領域を含む文字列領域を生成する文字列領域生成手段と、

前記文字列領域生成手段が生成した複数の文字列領域の属性に基づいて、該複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出するタイトル抽出手段とを備えることを特徴とするタイトル抽出装置。

【請求項２】　前記タイトル領域に含まれる文字領域を切り出して、文字認識を行う認識手段をさらに備えることを特徴とする請求項１記載のタイトル抽出装置。

【請求項３】　前記文字領域生成手段は、前記黒画素連結領域の外接矩形を前記文字領域として求め、前記文字列領域生成手段は、前記文書画像内における該外接矩形の辺の長さの頻度のヒストグラムを作成し、特定の頻度と該特定の頻度を与える最大の辺の長さとの関係表を作成し、該関係表を用いて、前記文書画像内において現れる頻度の高い第１の辺の長さを求め、該第１の辺の長さを基準として用いて前記文字列領域を生成することを特徴とする請求項１記載のタイトル抽出装置。

【請求項４】　前記文字列領域生成手段は、前記関係表において頻度が大きく変化する辺の長さをもとに、前記第１の辺の長さを決定することを特徴とす請求項３記載のタイトル抽出装置。

【請求項５】　前記文字列領域生成手段は、前記外接矩形の高さまたは幅を前記辺の長さとして用いて、前記ヒストグラムを作成することを特徴とす請求項３記載のタイトル抽出装置。

【請求項６】　前記文字列領域生成手段は、前記第１の辺の長さを用いて閾値を生成し、該閾値を用いて不要な文字領域を除去することを特徴とす請求項３記載のタイトル抽出装置。

【請求項７】　前記文字列領域生成手段は、前記閾値を用いて図表または写真の外接矩形を除去することを特徴とする請求項６記載のタイトル抽出装置。

【請求項８】　前記文字領域生成手段は、前記黒画素連結領域の外接矩形を前記文字領域として求め、前記文字列領域生成手段は、各外接矩形の第１の辺を用いて、前記文書領域内の第１の方向における外接矩形の分布範囲を表す第１のヒストグラムを作成し、該第１のヒストグラムの形状から外接矩形のグループを求め、該グループ毎に処理を行うことを特徴とす請求項１記載のタイトル抽出装置。

【請求項９】　前記文字列領域生成手段は、各外接矩形の第２の辺を用いて、前記文書領域内の第２の方向における外接矩形の分布範囲を表す第２のヒストグラムを作成し、前記第１および第２のヒストグラムの形状から前記グループを求め、該グループ毎に処理を行うことを特徴とする請求項８記載のタイトル抽出装置。

2

【請求項１０】　前記文字列領域生成手段は、前記第１の辺の中線上に頂点を持つ二等辺三角形を作成し、該二等辺三角形を用いて前記第１のヒストグラムを作成することを特徴とす請求項８記載のタイトル抽出装置。

【請求項１１】　前記文字列領域生成手段は、前記グループに属する外接矩形を探索して、重複する２つ以上の外接矩形を求め、該２つ以上の外接矩形を１つの外接矩形に統合することを特徴とす請求項８記載のタイトル抽出装置。

【請求項１２】　前記文字列領域生成手段は、前記グループに属する外接矩形を探索して、ネストしている外接矩形を求め、ネストを除去することを特徴とす請求項８記載のタイトル抽出装置。

【請求項１３】　前記文字列領域生成手段は、基準とする第１の外接矩形が属するグループ内の外接矩形を探索して、該第１の外接矩形に近接する第２の外接矩形を求め、該第１および第２の外接矩形の連結関係を表す連結関係表を作成し、該連結関係表を用いて前記文字列領域を生成することを特徴とす請求項８記載のタイトル抽出装置。

【請求項１４】　前記文字領域生成手段は、前記黒画素連結領域の外接矩形を前記文字領域として求め、前記文字列領域生成手段は、基準とする第１の外接矩形に近接する第２の外接矩形を求め、該第１および第２の外接矩形の連結関係を表す連結関係表を作成し、該連結関係表を用いて該第１および第２の外接矩形に同じ識別情報を付加することにより、該第１および第２の外接矩形を１つの文字列領域に統合することを特徴とす請求項１記載のタイトル抽出装置。

【請求項１５】　前記文字列領域生成手段は、前記第１の外接矩形から前記第２の外接矩形へ向かうポインタと、前記第２の外接矩形から前記第１の外接矩形へ向かうポインタのうち、少なくとも一方を前記連結関係表に格納することを特徴とす請求項１４記載のタイトル抽出装置。

【請求項１６】　前記文字列領域生成手段は、前記第１の外接矩形と前記第２の外接矩形の間に枠線がある場合には、該第１および第２の外接矩形を連結しないことを特徴とする請求項１４記載のタイトル抽出装置。

【請求項１７】　前記タイトル抽出手段は、

前記文字列領域の内部を横方向に複数の部分領域に分割して、各部分領域の中で黒画素占有率の大きな部分線分領域を抽出し、閾値以上の高さの部分線分領域については高さを無視して、横方向に連結している各部分線分領域を統合し、統合された線分領域を抽出する線分抽出手段を有し、

3

該線分領域を用いて前記タイトル領域を抽出することを特徴とする請求項1記載のタイトル抽出装置。

【請求項１８】　前記線分抽出手段は、前記文字列領域の内部を重複する複数の部分領域に分割することを特徴とする請求項１７記載のタイトル抽出装置。

【請求項１９】　前記線分抽出手段は、前記文字列領域の幅に近い長さの前記線分領域を抽出することを特徴とする請求項１７記載のタイトル抽出装置。

【請求項２０】　前記タイトル抽出手段は、前記線分領域が前記文字列領域内の下部にあるとき、該線分領域を下線と判別し、該文字列領域を前記タイトル領域の候補とすることを特徴とする請求項１７記載のタイトル抽出装置。

【請求項２１】　前記線分抽出手段は、前記文字列領域から同じ程度の左端座標および右端座標を持つ２つの線分領域を抽出し、該左端座標付近で縦方向の黒画素の第３のヒストグラムを作成し、該右端座標付近で縦方向の黒画素の第４のヒストグラムを作成し、第３および第４のヒストグラムのピークの高さが前記２つの線分領域の距離程度であれば、前記文字列領域内に枠線があると判別することを特徴とする請求項１７記載のタイトル抽出装置。

【請求項２２】　前記タイトル抽出手段は、前記複数の文字列領域の属性として下線属性または枠付き属性を抽出し、抽出した属性と各文字列領域の位置と文字列領域間の相対的位置関係とのうち少なくとも１つを用いて、各文字列領域にポイントを与え、高ポイントの文字列領域を前記特定の文字列領域とすることを特徴とす請求項１記載のタイトル抽出装置。

【請求項２３】　前記タイトル抽出手段は、下線属性または枠付き属性を持つ文字列領域に一定の得点を与えることを特徴とする請求項２２記載のタイトル抽出装置。

【請求項２４】　前記タイトル抽出手段は、第１の方向の中心座標が前記文書画像の中央付近にある文字列領域に一定の得点を与えることを特徴とす請求項２２記載のタイトル抽出装置。

【請求項２５】　前記タイトル抽出手段は、上下にある文字列領域との距離が離れている文字列領域に一定の得点を与えることを特徴とす請求項２２記載のタイトル抽出装置。

【請求項２６】　前記タイトル抽出手段は、左側に他の文字列領域がないような文字列領域に一定の得点を与えることを特徴とする請求項２２記載のタイトル抽出装置。

【請求項２７】　前記タイトル抽出手段は、枠線を含む第１の文字列領域の内部に第２の文字列領域があり、該第１の文字列領域と第２の文字列領域が閾値以上に離れていないような一定の位置関係にある場合に、該第２の文字列領域が枠付き属性を持つとみなすことを特徴とする請求項２２記載のタイトル抽出装置。

4

【請求項２８】　前記タイトル領域の位置またはサイズの情報から、他の文字列領域の相対的な位置関係またはサイズを求め、該他の文字列領域の相対的な位置関係またはサイズが特定の条件を満たすとき、該他の文字列領域を宛先領域として抽出する宛先抽出手段をさらに備えることを特徴とする請求項１記載のタイトル抽出装置。

【請求項２９】　前記タイトル領域の位置またはサイズの情報から、他の文字列領域の相対的な位置関係またはサイズを求め、該他の文字列領域の相対的な位置関係またはサイズが特定の条件を満たすとき、該他の文字列領域を発信元情報領域として抽出する発信元情報抽出手段をさらに備えることを特徴とす請求項１記載のタイトル抽出装置。

【請求項３０】　前記タイトル抽出手段は、前記文書画像内で一定範囲のサイズの文字列領域が存在する文書領域を求め、該文書領域から前記タイトル領域を抽出することを特徴とする請求項１記載のタイトル抽出装置。

【請求項３１】　前記タイトル抽出手段は、隣接した２つの文字列領域のサイズまたは座標値が類似している場合に、該２つの文字列領域を１つの文字列領域に統合することを特徴とする請求項１記載のタイトル抽出装置。

【請求項３２】　前記タイトル抽出手段は、文字列領域のサイズまたは形状から罫線を表すと判定した時、該文字列領域に罫線属性を設定し、該罫線属性を用いて前記タイトル領域を抽出することを特徴とす請求項１記載のタイトル抽出装置。

【請求項３３】　前記タイトル抽出手段は、前記罫線属性を持つ第３の文字列領域の上にある第４の文字列領域に下線属性を設定し、該第４の文字列領域を前記タイトル領域の候補とすることを特徴とす請求項３２記載のタイトル抽出装置。

【請求項３４】　文書を画像データに変換して得られる文書画像から必要とする部分領域を取り出して認識する情報処理装置において、
前記文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成する文字領域生成手段と、
前記文字領域生成手段が生成した１つ以上の文字領域を統合して、該１つ以上の文字領域を含む文字列領域を生成する文字列領域生成手段と、
一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、該表領域内の複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出するタイトル抽出手段とを備えることを特徴とするタイトル抽出装置。

【請求項３５】　前記タイトル抽出手段は、第５の文字列領域の内部に罫線がある場合に、該罫線の位置で該第５の文字列領域を分割することを特徴とす請求項３４記載のタイトル抽出装置。

【請求項３６】　前記タイトル抽出手段は、前記第５の文字列領域内の複数の文字領域の間に黒画素があるかどうかを調べ、黒画素がある位置で該第５の文字列領域を

分割することを特徴とする請求項３５記載のタイトル抽出装置。

【請求項３７】　前記タイトル抽出手段は、前記第５の文字列領域内の複数の文字領域と、前記文字領域生成手段が該第５の文字列領域内を対象にして再度求めた複数の文字領域との差異を調べ、該差異が検出された位置で該第５の文字列領域を分割することを特徴とす請求項３５記載のタイトル抽出装置。

【請求項３８】　前記タイトル抽出手段は、前記表領域内の第６の文字列領域の近くの特定領域に罫線があるかどうかを調べ、罫線がなければ該第６の文字列領域を表外の文字列領域とすることを特徴とす請求項３４記載のタイトル抽出装置。

【請求項３９】　前記タイトル抽出手段は、前記表領域内の文字列領域相互の位置関係から、上側に表内文字列領域がないような文字列領域を前記第６の文字列領域とし、該第６の文字列領域の上側にある前記特定領域の黒画素を探索し、一定閾値以上の黒画素が検出された場合に、該特定領域内に前記罫線があると判定することを特徴とする請求項３８記載のタイトル抽出装置。

【請求項４０】　前記タイトル抽出手段は、前記第６の文字列領域の上の他の文字列領域または表領域までの間を前記特定領域とし、前記閾値を該第６の文字列領域と該他の文字列領域または表領域との位置関係から決めることを特徴とす請求項３９記載のタイトル抽出装置。

【請求項４１】　前記タイトル抽出手段は、前記複数の文字列領域を前記表領域の左上に近い順に優先的に出力することを特徴とす請求項３４記載のタイトル抽出装置。

【請求項４２】　文字列領域生成手段は、前記１つ以上の文字領域を含む文字列矩形を文字列領域として生成し、前記タイトル抽出手段は、該文字列矩形の特定の頂点の座標値をもとに、前記表領域内の複数の文字列矩形に優先順位を付けることを特徴とす請求項４１記載のタイトル抽出装置。

【請求項４３】　前記タイトル抽出手段は、前記複数の文字列領域のうち、項目らしい文字列領域を項目領域とし、タイトルらしい文字列領域を前記タイトル領域として、優先順位を付けて出力することを特徴とす請求項３４記載のタイトル抽出装置。

【請求項４４】　前記タイトル抽出手段は、あらかじめ決められた項目とタイトルの位置および文字数の関係に該当する文字列領域のペアを求め、該文字列領域のペアを上から順に出力することを特徴とす請求項４３記載のタイトル抽出装置。

【請求項４５】　前記タイトル抽出手段は、前記複数の文字列領域のうち閾値以上の文字数を持つ第７の文字列領域を、前記項目領域として出力することを特徴とする請求項４３記載のタイトル抽出装置。

【請求項４６】　前記タイトル抽出手段は、前記第７の

文字列領域の右側の文字列領域を、前記タイトル領域として出力することを特徴とす請求項４５記載のタイトル抽出装置。

【請求項４７】　前記タイトル抽出手段は、前記複数の文字列領域のうち、閾値未満の文字数を持つ第８の文字列領域を、前記項目領域として出力し、該第８の文字列領域の右側にあって該閾値以上の文字数を持つ文字列領域を、前記タイトル領域として出力することを特徴とする請求項４３記載のタイトル抽出装置。

【請求項４８】　情報処理装置により用いられる記憶媒体であって、該情報処理装置が、

文書を画像データに変換して得られる文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成し、

１つ以上の文字領域を統合して、該１つ以上の文字領域を含む文字列領域を生成し、

複数の文字列領域の属性に基づいて、該複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出するように導くことを特徴とする記憶媒体。

【請求項４９】　情報処理装置により用いられる記憶媒体であって、該情報処理装置が、

文書を画像データに変換して得られる文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成し、

１つ以上の文字領域を統合して、該１つ以上の文字領域を含む文字列領域を生成し、

一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、

該表領域内の複数の文字列領域のうち、特定の文字列領域をタイトル領域として抽出するように導くことを特徴とする記憶媒体。

【請求項５０】　文書を画像データに変換して文書画像を生成し、

該文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成し、

１つ以上の文字領域を統合して、該１つ以上の文字領域を含む文字列領域を生成し、

複数の文字列領域の属性に基づいて、該複数の文字列領域のうち特定の文字列領域をタイトル領域として抽出し、

該タイトル領域に含まれる文字を認識することを特徴とするタイトル抽出方法。

【請求項５１】　文書を画像データに変換して文書画像を生成し、

該文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成し、

１つ以上の文字領域を統合して、該１つ以上の文字領域を含む文字列領域を生成し、

一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、

7

該表領域内の複数の文字列領域のうち、特定の文字列領域をタイトル領域として抽出し、
該タイトル領域に含まれる文字を認識することを特徴とするタイトル抽出方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は画像データの認識処理に係り、文書を画像データとして取り込んだ文書画像から、タイトル部分の領域を抽出するタイトル抽出装置および方法に関する。

【０００２】

【従来の技術とその問題点】一般文書をスキャナ等の光電変換装置で読み込んで得られる画像データである文書画像から、文書のタイトル等の部分領域を抽出する従来技術としては、以下に示す様なものがある。

（１）タイトルなどの領域が固定されている文書を対象として、固定領域をタイトルとして抽出する（特開昭６４－４６８７３）。

（２）文書に色マーカや枠線で囲むなどの特定のマークを付けてから、光電変換装置で読み込んで、特定の色部分や特定のマーク部分の抽出によってタイトル部分を抽出する（特開平１－１５０９７４）。

（３）文書の文字列や写真などの物理構造を木構造等に表現して、その論理構造とのマッチングをとることで、物理構造に「タイトル」、「著者名」等のタグ付けをする（特開平１－１８３７８４、特開平５－３４２３２６等）。

（４）文書画像の一部の領域を指定し、その内部を投影して黒画素のヒストグラムを作成する。そのヒストグラム上で、投影した黒画素の値が２つの閾値の間にある部分が連続する範囲を求め、その連続長が閾値より大きな部分をタイトルとして抽出する（特開平５－２７４４７１）。

【０００３】そのほかに、表を含む文書画像から表内のタイトル等の部分領域を抽出する従来技術として、以下に示す様なものがある。

（５）表を含む固定フォーマットの文書（タイトルなどの領域が固定されている文書）を対象として、固定された領域をタイトルとして抽出する（特開平７－０９３３４８）。

（６）文書画像を投影して黒画素のヒストグラムを作成し、ヒストグラムの分布から枠線を抽出して、枠線に囲まれる文字列をタイトルとして抽出する（特開平５－２７４３６７）。

（７）文書画像内の全文字領域を文字認識し、得られた文字コードに対して単語（キーワード）照合や形態素解析等の言語的、論理的な知識処理を行い、その結果からタイトルらしい文字列を抽出する（特開平３－２７６２６０）。

（８）文書画像内の白画素連結部分で囲まれた領域を表

8

部分として抽出し、その内部から罫線を抽出して、罫線で囲まれた領域を求める。そして、求めた領域の内部にある画像とあらかじめ決められた文字列（テンプレート）とのテンプレートマッチングを行うことで、それと同じ文字列をタイトルとして抽出する（特開平３－７４７２８）。

【０００４】しかしながら、これらの従来技術にはそれぞれ以下のような問題がある。

（１）および（５）の方法では、書式の固定した文書しか扱えない。書式を変更する場合は、抽出する部分の設定も変更する必要がある。

（２）の方法では、原稿文書にマークを付ける手間がかかる。

（３）の方法では、木構造等で表現した論理構造の辞書を用意する必要がある。また、辞書に無い論理構造の文書については、正確にタイトルを抽出することができなくなる。

（４）の方法では、文書画像の一部の領域の指定方法が明らかではないが、仮に全領域にこの方法を適用すると、図表などの大きな黒画素部分を誤ってタイトルとして抽出してしまう恐れがある。また文字だけの文書でも、文字サイズが大きな文字列がタイトルとは限らないので、誤抽出する可能性がある。

（６）単純な枠線に囲まれている表ならばこの方式でもよいが、実際には罫線が複雑に組み合わさった表が使われることが多いため、そのような場合にタイトル領域を正確に特定できない。

（７）現在の文字認識処理ではかなりの処理時間がかかるため、実質的にバッチ処理としてしか使用方法がない。また、認識率は１００％ではないので、タイトルの位置の情報を使わなければ、誤った部分をタイトルとして抽出することが多いと考えられる。

（８）画像上のテンプレートマッチングはマッチング処理自体に時間がかかるだけでなく、テンプレートのフォント形状またはサイズの影響を受けやすく、誤りやすいという欠点がある。また、この方法ではあらかじめ決まった文字列だけしかタイトルとして抽出できず、対象とする文書が限定される。

【０００５】このように、従来のタイトル抽出方法では、ユーザにとって特別な準備作業や操作が必要であったり、対象とする文書やタイトルが限定される等の問題がある。

【０００６】本発明は、文書画像から容易にタイトル部分を抽出することのできるタイトル抽出装置およびその方法を提供することを目的とする。

【０００７】

【問題を解決するための手段】図１は、本発明のタイトル抽出装置の原理図である。図１のタイトル抽出装置は、文字領域生成手段１、文字列領域生成手段２、およびタイトル抽出手段３を備える。

9

【０００８】文字領域生成手段１は、文書を画像データに変換して得られる文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成する。文字列領域生成手段２は、文字領域生成手段１が生成した１つ以上の文字領域を統合して、それらの文字領域を含む文字列領域を生成する。

【０００９】タイトル抽出手段３は、文字列領域生成手段２が生成した複数の文字列領域の属性に基づいて、それらの複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出する。

【００１０】文字領域生成手段１は文書画像内の黒画素を走査し、例えば、それらが連結している領域に外接する矩形領域を文字領域として抽出する。この結果、文書内の多数の文字に対応する多数の文字領域が生成される。

【００１１】次に、文字列領域生成手段２は、隣接する複数の文字領域を統合して、例えば、それらの文字領域に外接する矩形領域を文字列領域として抽出する。この文字列領域は、例えば、横書き文書内の１行分の文字列に対応する。

【００１２】タイトル抽出手段３は、生成された各文字列領域の下線属性、枠付き属性、罫線属性等の属性に基づいてタイトルらしさを評価し、最もタイトルらしいと考えられる特定の文字列領域を、タイトル領域として抽出する。

【００１３】ここで、下線属性とは、文字列領域の内部または下方に下線があることを意味し、下線フラグ等を用いて表現される。枠付き属性とは、文字列領域のまわりを枠線が囲んでいることを意味し、枠線フラグ等を用いて表現される。また、罫線属性とは、文字列領域が横長または縦長の罫線に対応することを意味し、罫線フラグ等を用いて表現される。下線属性や枠付き属性を持つ文字列領域は文書のタイトルである可能性が高く、罫線属性を持つ文字列領域はタイトルである可能性がほとんどない。そこで、このような属性をもとにタイトルらしさを自動的に評価することができる。

【００１４】また、タイトル抽出手段３は、一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、該表領域内の複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出する。

【００１５】表領域としては、例えば、黒画素連結領域に外接する矩形領域のうち、一定の閾値以上の大きさを持つものが用いられる。そして、タイトル抽出手段３は、表領域内の文字列領域相互の位置や文字数等の関係を調べて、タイトルらしさを評価し、最もタイトルらしいと考えられる特定の文字列領域を、タイトル領域として抽出する。

【００１６】例えば、表領域の左上に近い文字列領域ほどタイトルらしいと考えられ、また、文字数の大きな文字列領域もタイトルらしいと考えることができる。本発

10

明のタイトル抽出装置によれば、表形式文書を含む様々な文書画像を対象として、原稿にマークすることもなく、特別な構造辞書が不要で、文字サイズのみに影響されない、高精度なタイトル抽出処理を行うことができる。また、抽出されたタイトル領域に含まれる文字領域を切り出して文字認識を行い、認識結果を文書画像のキーワードとして用いることもできる。

【００１７】図１の文字領域生成手段１、文字列領域生成手段２、およびタイトル抽出手段３は、例えば、実施の形態における図２のプロセッサ１４に相当する。

【００１８】

【発明の実施の形態】以下、図面を参照しながら本発明の実施の形態を詳細に説明する。最近、従来より紙媒体で保存してきた情報を電子化する動きが多く見られる。その中の１つに電子ファイリングシステムがある。電子ファイリングシステムにおいては、紙文書がイメージスキャナ等の光電変換装置で画像に変換され、それに検索のためのキーワードや管理情報が付与されて、光ディスクやハードディスクに保存される。

【００１９】このような方法では文書を画像データとして保存するため、文書に記されているすべての文字を文字認識技術でコード化してから保存する方法よりも、ディスク容量は多く必要となる。その反面、手軽で処理速度が速く、文字以外の絵や表などもそのまま保存できるメリットがある。しかし、保存された情報を検索するために、文書画像と共にキーワードや番号などの管理情報を付与しなければならない。従来のシステムはこのキーワード付けの手間がかかり、使いづらいシステムであった。

【００２０】この使いづらさを解決するために、文書中にあるタイトル部分をキーワードとみなしてそれを自動的に抽出し、その部分を文字認識してコード化した結果を文書画像と共に保存する方法が考えられる。

【００２１】現在の文字認識の処理速度は速くても数十文字／秒であり、Ａ４の文書１枚を処理するのには３０秒から数分の処理時間がかかってしまう。したがって、タイトル抽出を高速化するためには、文書全体を文字認識するのではなく、まず画像上で必要なタイトル部分だけを抽出した後にそれを文字認識する方法が有力である。

【００２２】また、文書中の全文字を認識して論理的にタイトルを抽出する方法では、画像上におけるタイトル部分の位置関係が考慮されない。このため、誤認識や文章のつながりの影響で、正確にタイトルコードを抽出できない場合がかなりあるはずである。

【００２３】このように、電子ファイリングシステムの効率的な運用を考えると、文書画像から直接タイトル部分（領域）を抽出する技術は、重要な技術である。そこで、電子ファイリングシステムを例にとり、本発明のタイトル抽出技術について説明する。

11

【００２４】図２は、実施形態のタイトル抽出システムの構成図である。図２のタイトル抽出システムは、電子ファイリング装置１１、光電変換装置１２、ディスプレイ端末１３、プロセッサ１４、およびメモリ１５を備え、これらの装置はバス１６により結合されている。

【００２５】電子ファイリング装置１１は、ハードディスクや光ディスク等の格納装置を備え、複数の画像データを個別に格納している。光電変換装置１２は、例えばスキャナ等の光学的な読み取り装置であり、文書や絵、写真等を画像データに変換する。こうして取り込まれた画像データは、電子ファイリング装置１１またはメモリ１５に格納される。ディスプレイ端末１３は、ディスプレイ装置とキーボードやマウス等の入力装置とを備えたオペレータ端末である。

【００２６】プロセッサ１４は、ディスプレイ端末１３から入力された指示に従い、光電変換装置１２からメモリ１５に取り込んだ文書画像、または、電子ファイリング装置１１から取り出した文書画像からタイトル等の特定領域を抽出する。そして、抽出した領域に含まれる文字を認識する。尚、文字の認識処理は、タイトル抽出システムの外部のシステムにより行う構成としてもよい。

【００２７】図２のタイトル抽出システムは、例えば、後述する図４に示すような文書画像から図６に示すような文字の外接矩形を求め、さらに複数の文字の外接矩形を統合して、図２７に示すような文字列矩形を求める。そして、各文字列矩形が文書の中で強調されているかどうかを調べる。

【００２８】例えば、図１４に示すような枠線で囲まれている文字列は強調されているものとみなし、それだけでタイトルらしいと考えられるので、それらをタイトル候補として抽出する。そのほかにも、下線を持つ文字列や大きな文字列は強調文字列と考えて、タイトル候補として抽出する。また、文書内での文字列の位置や隣接する他の文字列との位置関係も、タイトル文字列を識別するために有力な情報として用いられる。

【００２９】このように、文字列が強調されているかどうかやその位置等の外見的な情報をもとにタイトル候補の文字列を選択するので、タイトルである蓋然性の高い領域を文書画像から簡単に抽出することができる。この抽出方法は文書全体を認識してからタイトルを抽出する方法よりも高速であり、対象とする文書を選ばないという点で汎用的である。また、文字列の２つ以上の外見的情報を組合せて用いることにより、比較的正確にタイトル領域を特定することができる。

【００３０】図３は、図２のタイトル抽出システムによるタイトル抽出処理のフローチャートを示している図３の処理においては、前提条件として横書き文書を対象としているが、横書きに限らず縦書き文書でも横書きと同様の処理で対応可能である。縦書き文書の場合には、文字領域や文字列領域の高さと幅が、横書き文書の場合と互いに逆の役割を果たすことになる。

12

【００３１】図３において処理が開始されると、光電変換装置１２が文書を読み取り、画像データ（文書画像）としてメモリ１５に格納する（ステップＳ１）。このとき、処理の高速化のために、読み取った原画像を縦横それぞれ１／８に圧縮して圧縮画像を作成し、それを文書画像としてメモリ１５に格納しておく。

【００３２】画像を圧縮する際には、線分がとぎれないように論理ＯＲの圧縮方法を用いる。つまり、原画像の８×８画素の領域の中に１つでも黒画素があれば、圧縮画像の対応する画素を黒とし、全く黒画素がなければそれを白とする。

【００３３】次に、プロセッサ１４が、文書画像から文字列（あるいは行）を抽出し、文字列の外接矩形（文字列矩形）を求め、その座標をメモリ１５に保存する（ステップＳ２）。次に、保存した文字列矩形から、横幅が小さい矩形や縦長矩形をノイズ矩形として除去し（ステップＳ３）、さらに文字列らしくない矩形を除いて、文書領域を決定する（ステップＳ４）。

【００３４】次に、残った文字列矩形を縦方向（ｙ座標）で並べ替え（ステップＳ５）、枠の画像を含む矩形（枠矩形）を抽出して、枠矩形内にある文字列矩形を枠付き矩形としてマークする（ステップＳ６）。また、下線の画像を含む矩形を抽出して、そのすぐ上にある文字列矩形を下線矩形としてマークする（ステップＳ７）。

【００３５】次に、タイトルらしさのポイント計算を行って、ポイントの高い文字列矩形をタイトルとして抽出し（ステップＳ８）、その結果を用いて文書の宛先と発信元情報を抽出する（ステップＳ９、Ｓ１０）。そして、抽出したタイトル、宛先、発信元情報の認識処理を行って（ステップＳ１１）、処理を終了する。

【００３６】次に、一般的な社内文書を例に取り、タイトル抽出処理を詳細に説明する。社内文書には、通常、「タイトル」、「宛先」、「発信日」、「発信所属」、「発信管理番号」、「本文（図表あり）」などの要素が含まれており、それらが様々な配置で記載されている。ここでは、このような様々な書式の文書からタイトル、宛先、および発信者情報（発信日、発信所属、発信管理番号等）を抽出する。

【００３７】図４は、スキャナでメモリ１５に読み込まれた文書画像の例を示している。図４の文書画像は、ソフトウェア販推レポートの送付表に関するものであり、この文書のタイトルは「ソフトウェア販推レポート　送付表」で、その下に宛先や発信元情報が記載されている。プロセッサ１４は、まずこの文書画像から文字列を抽出する。図５は、図３のステップＳ２の文字列抽出処理のフローチャートである。

【００３８】図５において処理が開始されると、プロセッサ１４は、まず文書画像から文字に相当する矩形を抽出する。そのために、文書画像に対してラベリングによ

る黒画素連結処理を施し、黒画素の外接矩形を求めて保存する（ステップＳ２１）。

【００３９】ここでは、２値化されている圧縮画像の黒画素を８連結で走査し、連結が有る場合にはそれらの黒画素に同一のラベル値を与えることによって黒画素連結領域を生成し、その外接矩形（文字矩形）を求める。８連結による走査とは、１つの黒画素の上、下、左、右、左上、右上、左下、右下の８方向を走査して、他の隣接黒画素があるかどうかを調べる処理を意味する。求められた外接矩形は、ファイルlbtbl に保存される図４の文書画像にラベリング処理を施した結果は図６のようになる。

【００４０】次に、ラベリングにより得られた外接矩形の高さの頻度分布を表すヒストグラムを求め、高さの最頻値freqを求める（ステップＳ２２）。ここでは、まずラベリング結果の外接矩形の集合lbtbl から図７に示すような矩形高さのヒストグラムを作成する。図７において、横軸が各外接矩形の高さを表し、縦軸がその高さを持つ矩形の数（頻度値）を表す。外接矩形の高さは、例えば１画素の高さを単位高さとして求める。

【００４１】次に、頻度値と、その頻度値を持つ矩形高さの内で最大の高さとの対応関係を求め、矩形高さテーブルheightに保存する。そして、heightの中を頻度値０から順に調査していき、高さの変化が１以内で頻度値の変化するものが連続し、それらの頻度値の変化の合計が９以上の場合に、それらの連続する高さのうちで最も高いものを矩形高さの最頻値freqとする。

【００４２】図８は、図７のヒストグラムに対応するheightの内容を表すヒストグラムを示している。図８において、頻度値が急激に変化する高さがfreqとなっていることがわかる。このようにしてfreqを求めておけば、１文字よりも小さなノイズの影響を排除することができる。

【００４３】図９は、heightの簡単な例を示している。図９においては、４つの頻度値と、各頻度値を持つ矩形高さの内で最大の高さとが、それぞれペアで格納されている。freqを求めるために、このheightの内容をヒストグラムにすると図１０のようになる。図１０のヒストグラムを、頻度値の低いところから、つまり高さの高いところから順に見ていくと、高さが１０、９、８の位置で、頻度値がそれぞれ５、５、７だけ変化していることが分かる。これらの連続する高さの差は１であり、頻度値の変化の合計は１７である。したがって、高さ１０、９、８において頻度値の変化の合計が９以上となっているので、それらの中で最初に現れた高さ１０をfreqとする。

【００４４】次に、枠線や図表の外接矩形を除去するために、大きな矩形と判断するための閾値を設定し、それより大きな矩形を抽出する。そして、抽出した大きな矩形の中から枠線を含む矩形を抽出して保存する（ステップＳ２３）。

【００４５】ここでは、例えば、freqより大きな矩形で最も頻度値が大きな矩形高さを大きな矩形の閾値th__large とし、th__large より大きな矩形を抽出して、ファイルbox に保存する。

【００４６】次に、box の中の大きな矩形から枠線を抽出するために、それぞれの大きな矩形の内部図１１に示すように縦に部分分割して、重複のある短冊状の部分領域を作る。そして、各短冊状部分領域の中で、一定割合以上の黒画素占有率を持つ高さ１画素の横方向の線状領域を求める。さらに２つ以上の線状領域が上下に連続していれば、それらを統合して１つにまとめた部分線分を求める。

【００４７】図１２は、図１１の大きな矩形の中の１つの短冊状部分領域を示している。図１２において、幅ｗの部分領域は高さ１の線状領域に細分され、一定割合以上の黒画素を含み、上下に連続する線状領域が１つの部分線分矩形として統合されている。図１２に示すように、１つの部分領域には２つ以上の部分線分矩形が存在する場合がある。次に、このようにして求めた部分線分矩形同士が左右で８連結の関係にあれば、それらを１つの線分として扱う。図１３（ａ）、（ｂ）、（ｃ）は、それぞれ８連結の関係にある２つの部分線分矩形の例を示している。こうして、図１１の場合は、大きな矩形の上端部分から横方向の線分矩形が抽出される。

【００４８】このようにして求めた線分矩形が、対象となる大きな矩形の横幅に比べて一定比率以上の場合に、長い線分矩形として抽出する。この長い線分矩形の両端と大きな矩形の両端の差が一定マージン以内にあり、かつ、長い線分矩形の上下端のｙ座標と大きな矩形の上下端のｙ座標の差が矩形幅の一定比率よりも小さい時に、大きな矩形の上下に位置している横罫線と判断する。

【００４９】そして、この大きな矩形の左右端付近の黒画素を縦方向に投影した頻度分布（ヒストグラム）を求め、そのピークの高さが矩形高さの一定比率より大きい場合に、左右端に縦罫線もあると判断する。このとき、この大きな矩形は枠線の外接矩形（枠矩形）と識別される。box の中の各大きな矩形について同様の処理を行い、枠矩形のみをbox に残す。図１４は、検出された枠矩形を示している。

【００５０】次に、ラベリングで求めた外接矩形の集合lbtbl から枠矩形および図表と判定された矩形（図表矩形）を除去して、結果を保存する（ステップＳ２４）。ここでは、まずlbtbl の中からbox に保存されている枠矩形を除去する。さらに、次のいずれかに該当する矩形を図表矩形と推定して、これらをlbtbl から除去する。

（ａ）文書画像全体の高さの１／３より大きな矩形

（ｂ）高さがfreqの３倍より大きく、高さ／幅の比が０．４より小さい矩形

（ｃ）高さがfreqの３倍より大きく、文書画像全体の幅

の１／３より大きな矩形

そして、除去後の矩形集合をnewtblとして管理する。この矩形集合newtblから文字列の外接矩形が抽出される。

【００５１】newtbl内の矩形の中には、矩形同士が重複またはネストしているものも含まれている。このような矩形は１つにまとめた方が、矩形相互の位置関係が明確になり、文字列の抽出を効率的に行うことができる。そこで、newtblを対象にして、矩形同士が重複またはネストしているものを統一して、重複／ネストを除去し、結果をファイルlbtbl2に保存する（ステップＳ２５）。

【００５２】図１５は、重複する２つの矩形の例を示している。図１５において、矩形２１と矩形２２は、それぞれ右上がりの斜めの線分の外接矩形を表し、斜線部分で互いに重複している。この場合、矩形２１、矩形２２を、これらの矩形を包括する１つの矩形２３にまとめて、重複を除去する。図１６は、ネストしている複数の矩形を示している。図１６において、矩形２５、２６、２７は、矩形２４に完全に内包されており、その矩形にネストしている。この場合、矩形２４のみを残して、ネストを除去する。

【００５３】ところで、newtblの中で、１つの矩形と重複／ネストしている他の矩形を探索するのには、次の２つの方法がある。

（ｄ）１つの矩形を基準として、残りの矩形全部を探索範囲とする。

（ｅ）縦または横方向に、矩形の辺の中線上に頂点を持つ二等辺三角形を作り、そのヒストグラムを作成する。そして、ヒストグラム中のそれぞれの頻度の山を構成する矩形の集合（グループ）を同時に記録する。ヒストグラム中で、山と山の距離が閾値より近いものを統合し、同時に対応する矩形集合も統合する。この矩形の集合を１つの探索範囲とし、この集合内にある矩形を基準とした場合は、この集合内を探索する。また、縦方向および横方向で作成した各矩形集合の重なり部分を求めて、探索範囲としてもよい。

【００５４】図１７は、（ｅ）の方法で用いる二等辺三角形のヒストグラムの例を示している。図１７において、矩形３１、３２の二等辺三角形３６、３７が１つの山４１に投影され、矩形３３の二等辺三角形３８が山４２に投影され、矩形３４、３５の二等辺三角形３９、４００が山４３に投影されている。例えば、これらの山４１、４２、４３が一定距離内にある場合は、矩形３１、３２、３３、３４、３５は１つの矩形集合に統合される。あるいはまた、矩形３１と矩形３２のように、対応する二等辺三角形が１つの山に投影されるような矩形を１つの矩形集合にまとめてもよい。

【００５５】（ｅ）の方法によれば、限られた範囲の矩形のみを探索すればよいので、一般に（ｄ）の方法よりも高速処理が可能である。図１８は、このようにして重複／ネストが除去された外接矩形を示している。

【００５６】次に、重複／ネスト除去後のlbtbl2に含まれる矩形の高さのヒストグラムを求め、高さの最頻値freq2 を求める（ステップＳ２６）。高さのヒストグラムの作成方法およびfreq2 を求める方法は、ステップＳ２２と同様である。

【００５７】次に、lbtbl2から罫線矩形を抽出して、マークする（ステップＳ２７）。ここでは、lbtbl2内で、高さがfreqの１／２より小さく、幅がfreqの３倍より大きく、高さ／幅の比が０．１より小さい矩形を、罫線矩形としてマークする。

【００５８】次に、１つの文字列に属する複数の文字を見つけるために、lbtbl2内の矩形相互の関係を求め、連結関係表connect に保存する（ステップＳ２８）。ここでは、lbtbl2内の各矩形から上下左右に最も近い矩形を探索し、その結果をconnectに格納する。矩形相互の関係とは、ある基準矩形から上下左右の矩形への各ポインタと、上下左右の矩形から基準矩形へ向かう各ポインタ、および基準矩形から上下左右の矩形までの距離を意味する。

【００５９】図１９は、１つの矩形を基準矩形とした場合の矩形間の連結関係を示している。図１９において、上矩形は基準矩形の上に近接する矩形を表し、ポインタ５１、５２により基準矩形と連結されている。上矩形は基準矩形の上に近接する矩形を表し、ポインタ５１、５２により基準矩形と連結されている。下矩形は基準矩形の下に近接する矩形を表し、ポインタ５３、５４により基準矩形と連結されている。左矩形は基準矩形の左に近接する矩形を表し、ポインタ５５、５６により基準矩形と連結されている。右矩形は基準矩形の右に近接する矩形を表し、ポインタ５７、５８により基準矩形と連結されている。

【００６０】このようなポインタを格納する連結関係表connect の構造は、例えば図２０に示すようになる。図２０の連結関係表には、基準矩形のラベル値に続いて、上矩形へのポインタ、上矩形からのポインタ、下矩形へのポインタ、下矩形からのポインタ、左矩形へのポインタ、左矩形からのポインタ、右矩形へのポインタ、右矩形からのポインタが格納されている。connect には、これらのポインタの他に基準矩形から上下左右の各矩形までの距離も格納される。

【００６１】connect を作成する時には、枠矩形の４辺で連結関係が切断されるように設定しておく。これは、後に枠線を越えて文字列を抽出しないようにするためである。基準矩形に最も近い矩形を探索する場合も、ステップＳ２５で用いた（ｄ）と（ｅ）の２通りの方法がある。

【００６２】次に、スキャナによる読み取り時のノイズに相当するノイズ矩形を識別し、それと他の矩形との横の関係を切断する（ステップＳ２９）。ここでは、矩形高さ、幅がfreq2 の１／４より小さい矩形、または、高

17

／幅の比が０．１より小さいか１０より大きくかつ上下の矩形との距離が一定値より大きい矩形を、ノイズ矩形と判定する。そして、その矩形と他の矩形との間の横方向のポインタを削除して、連結関係を切断する。

【００６３】次に、隣接する矩形間の距離が離れている場合、または隣接する矩形間で大きさに差がある場合、それらの矩形の連結関係を切断する（ステップＳ３０）。ここでは、基準矩形が次のいずれかの条件に該当する場合に、隣の矩形との連結関係を切断する。

（ｆ）基準矩形と隣の矩形との距離が、freq2 の３倍よりも大きい。

（ｇ）基準矩形または隣の矩形が、freq2 の３倍以上の大きさを持つ。

（ｈ）隣の矩形がfreq2 の２倍より大きい。

【００６４】次に、文字矩形の集合lbtbl2とその連結関係表connect から、文字列を抽出し、文字列の外接矩形（文字列矩形）を保存する（ステップＳ３１）。ここでは、まず、lbtbl2内の矩形のうち、その矩形へ左から向かうポインタが無いもの、つまり、左側に矩形が無いものを開始矩形とする。次に、その矩形の識別番号（例え20ばラベル値）を右側にある他の矩形へ順次伝搬させていき、同じ識別番号を付加した複数の矩形を統合して、それらの外接矩形を文字列矩形とする。この時、開始矩形の識別番号を、抽出した文字列の識別番号（ラベル値）としてline__lab に保存しておく。そして、右側に連結する矩形が無くなったら伝搬を終了する。

【００６５】図２１は、こうして抽出された文字列矩形の例を示している。図２１では、横に並んだ４つの文字矩形がラベル値Ｌ１を付加されて、１つの文字列矩形に統合されている。この場合、その文字列矩形のラベル値30もＬ１になる。

【００６６】もし、右側の矩形の識別番号が、既にline__lab 内にある文字列識別番号に一致する場合は、これまで伝搬してきた矩形の集合の識別番号を、右側の文字列識別番号へ置き換える。そして、置き換え前の文字列の識別番号はline__lab から除去しておく。

【００６７】この処理の後、左から向かうポインタが無い矩形を再び検出し、これを基準矩形とする。その基準矩形の左側に矩形がある場合は、その左側の矩形は既に抽出された文字列の識別番号に組み込まれているはずで40ある。そこで、その番号を基準矩形より右にある矩形に対して、右に連結した矩形が無くなるまで伝搬させ、識別番号を置き換える。そして、line__lab から置き換え前の矩形の番号を除去する。

【００６８】例えば、図２２に示すように、１つの文字列矩形の中に他の文字列矩形が存在する場合を考える。左からのポインタを持たない矩形６４を基準矩形として、その左へのポインタを辿ると左側に矩形６１があることが分かる。矩形６１は既にラベル値Ｌ０を持っているので、この値を矩形６４、６５へ伝搬させて、それら50

18

のラベル値をＬ０に置き換える。こうして、ラベル値Ｌ５はline__lab から除去され、矩形６１、６２、６３、６４、６５は１つの文字列矩形に統合される。

【００６９】ここまでの処理において、同一文字列と識別された各矩形には同じ文字列識別番号がついている。そこで、全部の矩形を走査して、同じ文字列識別番号が付いている複数の矩形の座標から最左端、最右端、最上端、最下端を求め、それらを文字列矩形の外周を構成する座標として、ファイルlineに保存する。また、抽出した文字列の数をmaxline として保存する。

【００７０】以上で、文字列抽出処理が終了する。図２３は、こうして抽出された文字列矩形を示している。次に、プロセッサ１４は、抽出された文字列矩形に対して、図３のステップＳ３からＳ７までの処理に対応する文字列矩形加工処理を施す。文字列矩形加工処理においては、各文字列矩形の下線属性、枠付き属性、罫線属性等の属性を抽出し、それらを記録する。後のポイント計算において、下線属性や枠付き属性を持つ文字列矩形にはより高いポイントが付与され、罫線属性を持つ文字列矩形にはより低いポイントが付与される。

【００７１】図２４は、この文字列矩形加工処理のフローチャートである。図２４において処理が開始されると、まず横幅が小さい文字列矩形や縦長の文字列矩形をノイズ文字列矩形として除去し、結果を保存する（ステップＳ４１）。ここでは、横幅がfreq／４より小さい文字列矩形、または、高さがfreq／４より小さくかつ高さ／幅の比が０．１より大きい文字列矩形をノイズとみなして除去し、残ったものをファイルline2 に保存する。図２５は、ノイズ除去後の文字列矩形を示している。

【００７２】次に、line2 内の文字列矩形間の接続関係を表す連結関係表 str__connを作る（ステップＳ４２）。ここでの接続関係は、図１９に示した文字矩形間の連結関係と同様のものを表し、 str__conn図２０に示した連結関係表と同様の構造を持つ。

【００７３】次に、位置関係や高さが一定の条件を満たす２つ以上の文字列を統合してより長い文字列を求め、結果を保存する（ステップＳ４３）。ここでは、次のいずれかに該当する場合に、それらの文字列矩形を１つに統合し、さらに大きな文字列矩形を求める。

（ｉ）文字列矩形間の距離が文字列矩形の高さより小さい場合

（ｊ）横方向に重複していて、高さがほぼ等しい文字列矩形

（ｋ）矩形高さの最頻値freq位の高さで、他の文字列矩形に完全に含まれる文字列矩形

（ｌ）３連の文字列矩形で両端の矩形のｙ座標がほぼ等しく、それらの間にある矩形だけが異なる場合

図２６（ａ）、（ｂ）、（ｃ）、（ｄ）は、それぞれ（ｉ）、（ｊ）、（ｋ）、（ｌ）の場合に統合されてできる文字列矩形の例を示している。このような処理を文

字列矩形の数が変化しなくなるまで繰り返し、残った文字列矩形をファイルline3 に保存する。図２７は、こうして文字列矩形を統合した結果を示している。図２５と図２７を比べると、例えば、文字列矩形「ソフトウェア販推レポート」と「送付表」とが、文字列矩形「ソフトウェア販推レポート　送付表」に統合されていることが分かる。

【００７４】次に、文字列の高さのヒストグラムを作成し、文字列高さの最頻値 str__freqを求める（ステップＳ４４）。ここでは、文字列矩形の高さのヒストグラム10を、図７と同様にして作成する。そのヒストグラムから、freq2 以上で最大頻度を与える高さを求め、それを文字列矩形の高さの最頻値 str__freqとする。もし、最大頻度を与える高さが複数個あったら、freq2 に近い方の高さを採用する。文字列矩形の高さのヒストグラムにおいて、 str__freqから連続する頻度分布を見ていったとき、 str__freqの両側に頻度値が０になる位置がある。これらの頻度値が０になる位置の直前の高さのうち、小さい方をst__h 、大きい方をen__h とする。

【００７５】次に、ノイズを除いた文書領域を求めて、その領域の座標を保存する（ステップＳ４５）。ここでは、文書画像の左右端にある一定領域内に一部分でも掛かるような文字列矩形は対象外として、高さがst__h 以上、en__h 以下で、かつ、横幅が str__freq以上で、かつ、高さ／幅の比が０．５未満の文字列矩形が存在する範囲を文書領域とする。そして、その領域の左端のｘ座標、上端のｙ座標、右端のｘ座標、下端のｙ座標を、それぞれst__x 、st__y 、en__x 、en__y として保存する。左右端の一定領域を無視するのは、例えばＡ４サイズの画像領域にＢ５版の本の１ページ分の画像を読み込んだような場合に、文書画像の左右に存在する隣のページの文字列矩形を、ノイズとして除去するためである。図２８は、こうして求められた文書領域を示している。

【００７６】次に、line3 内の文字列矩形を縦方向（ｙ座標）で並べ替える（ステップＳ４６）。次に、line3 内の文字列矩形間の連結関係を表す連結関係表 str__conn2 を作る（ステップＳ４７）。この時、枠矩形を跨いで連結する関係がないようにする。

【００７７】次に、各文字列矩形が枠矩形に完全に含まれているかどうかをチェックし、含まれている場合にはその文字列矩形に枠付きフラグを立てる（ステップＳ４８）。ここでは、line3 内の各文字列矩形に対して、それがbox に保存された枠矩形に完全に内包される場合に枠付き矩形とみなし、その文字列矩形に枠付きフラグを立てる。枠付き矩形の判定基準としては、枠矩形の内部にある文字列矩形をすべて枠付き矩形とみなす場合と、枠矩形と内部の文字列矩形の座標値が閾値以上に離れていない場合のみ枠付き矩形とみなす場合とがある。

【００７８】次に、line3 内の文字列矩形の中で、罫線矩形と判断したものに罫線フラグを立てる（ステップS50

４９）。ここでは、 str__freqの１／２以下で、高さ／幅の比が０．８より小さいか、または１２．５より大きいものを罫線矩形とみなして、その文字列矩形に罫線フラグを立てる。

【００７９】次に、line3 内の文字列矩形を調べてその直下に下線らしい罫線矩形（下線矩形）が有る場合、または、文字列矩形内部を走査して内部に下線が有る場合は、その文字列矩形に下線フラグを立てる（ステップＳ５０）。ここでは、罫線矩形の上に文字列矩形があり、それらの間の距離が str__freqより小さい範囲にあり、かつ、上の文字列矩形と罫線矩形の左右端の差が str__freq以下のとき、上にある文字列矩形に下線フラグを立てる。図２９は、下線矩形の例を示している。図２９において、文字列矩形７１の下には罫線フラグが立てられた横長の罫線矩形７２があるため、これが下線矩形とみなされ、文字列矩形７１には下線フラグが立てられる。

【００８０】また、幅または高さが str__freqの１／２以上の文字列矩形を対象として、後に述べる方法で線分を抽出する。そして、文字列矩形内で抽出した線分が、文字列矩形の左右端から一定画素数の範囲にあり、かつ、線分の高さが矩形高さのWAKUTHIN倍（例えば０．３倍）以下で、かつ、線分の下のｙ座標が矩形の下のｙ座標から str__freq／２だけ上の位置より下にあり、かつ、線分の上のｙ座標と矩形の上のｙ座標の差が str__freq－２よりも大きく、かつ、線分の下のｙ座標と矩形の下のｙ座標の差が線分の上のｙ座標と矩形の上のｙ座標の差よりも小さい場合に、この線分を文字列矩形内の下線として識別し、その文字列矩形に下線フラグを立てる。

【００８１】こうして、文字列加工処理を終了する。図３０は、枠付きフラグ、罫線フラグ、下線フラグを立てる処理を終えた後の文字列矩形を示している。図３０において、Ｌ０～Ｌ５４は、各文字列矩形に付加されたラベル値を表している。これらの文字列矩形のうち、ラベル値Ｌ１、Ｌ２、Ｌ１６を持つ文字列矩形が枠付き矩形に相当する。

【００８２】次に、図２４のステップＳ５０で文字列矩形から線分を抽出する方法を詳細に説明する。図３１は、線分抽出処理のフローチャートである。図３１において処理が開始されると、プロセッサ１４は、まず文字列矩形を一定画素幅ｗの短冊状の部分領域に分割する（ステップＳ６１）。この部分領域は、図１１の場合と同様に半分ずつ重なるような領域とする。

【００８３】次に、各部分領域の内部を上から下へ順に、縦１画素×横ｗ画素の線状領域毎に注目していく。ある線状領域の内部の黒画素数が閾値よりも大きい場合に、この線状領域の内部が全て黒画素であるとみなし、これを黒領域とする。黒領域の直下に別の黒領域がある場合は、２つの黒領域は連続しているものと判断し、１つの黒領域（部分線分矩形）として扱う（ステップＳ６

２）。すなわち、黒領域を表す座標は、左右は部分領域の左右の座標、上は、上から順に走査していったときに白領域から黒領域へ変化するときの黒領域のｙ座標、下は、黒領域から白領域に変化するときの黒領域のｙ座標となる。この結果、１つの部分領域から１つあるいは複数個の黒領域の座標が求められる。この操作を全部分領域で行い、黒領域の集合を求める。

【００８４】次に、黒領域の中で高さが閾値より大きいものをワイルドカードと呼ぶことにする（ステップＳ６３）。ワイルドカードは、例えば、文字列矩形内で文字10が潰れて黒画素の塊になっているような場合に発生する。図３２は、部分領域に分割された文字列矩形と、その中のワイルドカードの例を示している。また図３３は、１つの部分領域の中の線状領域とワイルドカードの例を示している。図３３において、部分領域は１５個の線状領域からなり、それらのうち上から１２個の線状領域がワイルドカードを形成している。

【００８５】次に、黒領域の集合を走査し、重複または隣接するものを統合して横長の矩形領域を求める（ステップＳ６４～Ｓ６９）。まず、最初に黒領域の集合から20１つの黒領域を選び、それに注目する（ステップＳ６４）。その黒領域がワイルドカード矩形でない場合は、その黒領域の上下端の座標と左右端の座標を、横長の矩形領域の座標として保存する。１回でも黒領域の集合から取り出した黒領域は、使用済みフラグを立てて二度と使用しない。

【００８６】次に、黒領域の集合から１つの黒領域を取り出し、既に使用済みのものでなければ、記憶した横長矩形の座標と比較して、その右側に隣接または重複する関係にあるかどうかをチェックし、そのような関係にあ30る黒領域を選ぶ（ステップＳ６５）。そして、その黒領域がワイルドカードかどうかを判定し（ステップＳ６６）、ワイルドカードの場合はその高さを無視して横方向に領域を統合する（ステップＳ６７）。このとき、記憶している横長矩形の右端の座標を、そのワイルドカード矩形の右端の座標で置き換える。

【００８７】右側に隣接または重複する黒領域がワイルドカードでない場合は、両方の矩形の上下座標を比較し、それらの差が閾値以内にあれば、縦方向と横方向に領域を統合する（ステップＳ６８）。このとき、右側の40ワイルドカードでない黒領域の上下座標を新しい横長の矩形領域の上下座標とする。また、黒領域の右端の座標を横長矩形の右端の座標とする。そして、黒領域をすべて調べたかどうか判定し（ステップＳ６９）、未処理の黒領域があれば、ステップＳ６５以降の処理を繰り返す。さらに、注目する黒領域を他のものに変えて（ステップＳ７０、Ｎｏ）、ステップＳ６４以降の処理を繰り返し、すべての黒領域を取り出すと処理を終了する。

【００８８】このように、図３１の線分抽出処理においては、まず矩形内部を適当な長さの重複がある縦短冊に50

分割し、１つの短冊内部で一定の黒画素占有率を満たす部分を抽出して部分線分矩形（黒領域）で表現し、それらを保存する。ここまでは、図１１に示した線分の抽出方法と同じである。このとき、保存された部分線分矩形は、下線の一部である高さの小さい矩形の場合もあるが、文字が潰れてそれが下線と接触しているときには、図３２のワイルドカードのような高さの大きな矩形の場合もある。これらを横方向に走査していき、全体的な１つの長い線分矩形として抽出する。図３２においては、文字列矩形内のワイルドカードの高さは無視されて、他の部分線分矩形と統合され、文字列矩形の下端部分に横長の線分矩形が抽出されている。

【００８９】図３４、３５、３６は、線分抽出処理のプログラムコードの例を示している。図３５は、図３４のＣ１の位置のαに相当する部分を示しており図３６は、図３４のＣ２の位置のβに相当する部分を示している。また、図３７、３８、３９は、図３４、３５、３６の処理の概要を示すフローチャートである。この処理においては、文字が潰れてできた大きな黒画素塊をワイルドカード矩形として扱い、その前後に８連結で接続される横長の矩形に注目する。そして、ワイルドカード矩形を挟んでお互いに８連結の関係にある矩形を統合していき、１つの横に長い矩形を線分候補の罫線として求める。以下、図３７、３８、３９を参照しながら、具体的な処理を説明する。

【００９０】図３７において処理が開始されると、プロセッサ１４は、まず各部分線分矩形の高さを調べる（ステップＳ７１）。そして、それが文字列矩形の高さ×０．３以上であれば、ワイルドカード矩形としてマークする（ステップＳ７２）。このとき、その部分線分矩形の識別変数ｕｓｅを９とおくことにより、ワイルドカード印をつける。それ以外の部分線分矩形は普通の矩形（スタンダード矩形）として、ｕｓｅ＝０とおく（ステップＳ７３）。そして、すべての部分線分矩形をマークしたかどうかを判定し（ステップＳ７４）、まだ部分線分矩形が残っていれば、ステップＳ７１以降の処理を繰り返す。

【００９１】すべての部分線分矩形をマークし終えると、１つの矩形をカレント矩形ｉとして取り出し、ｘｌｆ＝カレント矩形ｉの左端座標、ｘｒ＝カレント矩形ｉの右端座標、ｙｕｐ＝カレント矩形ｉの上端座標、ｙｂｌ＝カレント矩形ｉの下端座標、ｌｉｎｅ＿ｓｔａｒｔｙ＝ｙｕｐ、ｌｉｎｅ＿ｅｎｄｙ＝ｙｂｌとおく（ステップＳ７５）。そして、カレント矩形ｉのｕｓｅが０または９であるかどうか調べる（ステップＳ７６）。

【００９２】カレント矩形ｉのｕｓｅが０または９であれば、次にｕｓｅ＝０かどうかを判定する（ステップＳ７７）。ｕｓｅ＝０であれば、ｓｔａｎｄａｒｄ＿ｓｔ＝ｙｕｐ、ｓｔａｎｄａｒｄ＿ｅｎ＝ｙｂｌ、ｂ＿ｕｓｅ＝０、ｕｓｅ＝１、ｈｅｉｇｈｔ＝ｙｂｌ－ｙｕｐ＋

１とおく（ステップＳ７８）。ｂ＿ｕｓｅ＝０は、カレント矩形ｉがワイルドカードではなく、スタンダードとして設定されていることを意味し、ｕｓｅ＝１はカレント矩形ｉが使用済みであることを意味する。ステップＳ７６でｕｓｅ＝０でなければ、ｓｔａｎｄａｒｄ＿ｓｔ＝０、ｓｔａｎｄａｒｄ＿ｅｎ＝０、ｂ＿ｕｓｅ＝９、ｈｅｉｇｈｔ２＝ｙｂｌ－ｙｕｐ＋１とおく（ステップＳ７９）。ｂ＿ｕｓｅ＝９は、カレント矩形ｉがワイルドカードであるため、スタンダードとして設定されないことを意味する。

【００９３】次に、他の部分線分矩形をカレント矩形ｋとして取り出し、ｒｘｌｆ＝カレント矩形ｋの左端座標、ｒｘｒ＝カレント矩形ｋの右端座標、ｒｙｕｐ＝カレント矩形ｋの上端座標、ｒｙｂｌ＝カレント矩形ｋの下端座標とおく 図３８、ステップＳ８０）。そして、カレント矩形ｉがスタンダードとして設定されているかどうか、すなわち、ｂ＿ｕｓｅ＝０であるかどうかを調べる（ステップＳ８１）。ｂ＿ｕｓｅ＝０であれば、次に、カレント矩形ｋのｕｓｅが９であるかどうかを調べる（ステップＳ８２）。ここで、ｕｓｅ＝９の場合は、カレント矩形ｉがスタンダードで、カレント矩形ｋがワイルドカードであることを意味をする。

【００９４】ｕｓｅ＝９のとき、ｘｒ＋１≧ｒｘｌｆ、ｘｒ＜ｒｘｒ、ｙｂｌ＋１≧ｒｙｕｐ、およびｙｕｐ－１≦ｒｙｂｌが成り立つかどうかを判定する（ステップＳ８３）。これらが成り立つ時、カレント矩形ｋがカレント矩形ｉの右側にあり、両者が横と縦に１画素（１ドット）以上の重なりを有することを意味する。これらの条件が成り立つ時、ｘｒ＝ｒｘｒとおいて、カレント矩形ｉの右端をカレント矩形ｋの右端まで延長する（ステップＳ８４）。

【００９５】ステップＳ８２でｕｓｅ＝９でないとき、次に、ｕｓｅ＝０であるかどうかを調べる（ステップＳ８５）。ここで、ｕｓｅ＝０の場合は、カレント矩形ｉがスタンダードで、カレント矩形ｋがワイルドカードでないことを意味をする。ｕｓｅ＝０のとき、ｘｒ＋１≧ｒｘｌｆ、ｘｒ＜ｒｘｒ、ｙｂｌ＋１≧ｒｙｕｐ、およびｙｕｐ－１≦ｒｙｂｌが成り立ち、かつ、カレント矩形ｋの高さが一定範囲内かどうかを判定する（ステップＳ８６）。

【００９６】これらの条件が成り立つ時、ｘｒ＝ｒｘｒ、ｙｕｐ＝ｒｙｕｐ、ｙｂｌ＝ｒｙｂｌ、ｕｓｅ＝２、ｈｅｉｇｈｔ＝ｒｙｂｌ－ｒｙｕｐ＋１とおく（ステップＳ８７）。これは、カレント矩形ｉの右端をカレント矩形ｋの右端まで延長し、上下端の座標をカレント矩形ｋのものに置き換えることを意味する。ここで、ｕｓｅ＝２はカレント矩形ｋが使用済みであることを意味する。次に、ｒｙｕｐ＜ｌｉｎｅ＿ｓｔａｒｔｙが成り立つかどうかを判定し（ステップＳ８８）、成り立てばｌｉｎｅ＿ｓｔａｒｔｙ＝ｒｙｕｐとおく（ステップS50

８９）。さらに、ｒｙｂｌ＞ｌｉｎｅ＿ｅｎｄｙが成り立つかどうかを判定し（ステップＳ９０）、成り立てばｌｉｎｅ＿ｅｎｄｙ＝ｒｙｂｌとおく（ステップＳ９１）。

【００９７】これらの処理の後、次にｂ＿ｕｓｅ＝９かどうかを判定する 図３９、ステップＳ９２）。ステップＳ８１でｂ＿ｕｓｅ＝０でないとき、あるいはステップＳ８３、Ｓ８５、Ｓ８６、Ｓ８８、Ｓ９０で判定結果がＮｏのときは、直ちにステップＳ９２以降の処理に移る。

【００９８】ｂ＿ｕｓｅ＝９であれば、次に、カレント矩形ｋのｕｓｅが９であるかどうかを調べる（ステップＳ９３）。ここで、ｕｓｅ＝９の場合は、カレント矩形ｉとカレント矩形ｋの両方がワイルドカードであることを意味をする。ｕｓｅ＝９であれば、ｘｒ＋１≧ｒｘｌｆおよびｘｒ＜ｒｘｒが成り立つかどうかを判定する（ステップＳ９４）。これらが成り立つ時、カレント矩形ｋがカレント矩形ｉの右側にあり、両者が横と縦に１ドット以上の重なりを有するので、ｘｒ＝ｒｘｒとおいて、カレント矩形ｉの右端をカレント矩形ｋの右端まで延長する（ステップＳ９５）。

【００９９】ステップＳ９３でｕｓｅ＝９でないとき、次に、ｕｓｅ＝０であるかどうかを調べる（ステップＳ９６）。ここで、ｕｓｅ＝０の場合は、カレント矩形ｉがワイルドカードで、カレント矩形ｋがワイルドカードでないことを意味をする。ｕｓｅ＝０のとき、ｘｒ＋１≧ｒｘｌｆおよびｘｒ＜ｒｘｒが成り立つかどうかを判定する（ステップＳ９７）。これらの条件が成り立つ時、ｘｒ＝ｒｘｒ、ｙｕｐ＝ｒｙｕｐ、ｙｂｌ＝ｒｙｂｌ、ｕｓｅ＝２、ｌｉｎｅ＿ｓｔａｒｔｙ＝ｒｙｕｐ、ｌｉｎｅ＿ｅｎｄｙ＝ｒｙｂｌ、ｈｅｉｇｈｔ＝ｒｙｂｌ－ｒｙｕｐ＋１、ｓｔａｎｄａｒｄ＿ｓｔ＝ｒｙｕｐ、ｓｔａｎｄａｒｄ＿ｅｎ＝ｒｙｂｌとおく（ステップＳ９８）。これは、カレント矩形ｉの右端をカレント矩形ｋの右端まで延長し、上下端の座標をカレント矩形ｋのものに置き換えることを意味する。また、ｕｓｅ＝２はカレント矩形ｋが使用済みであることを意味する。

【０１００】次に、カレント矩形ｋとしてすべての部分線分矩形を取り出したかどうかを判定する（ステップＳ９９）。ステップＳ９２でｂ＿ｕｓｅ＝９でないとき、あるいはステップＳ９４、Ｓ９６、Ｓ９７で判定結果がＮｏのときは、直ちにステップＳ９９以降の処理に移る。ステップＳ９９で、残っている部分線分矩形があればステップＳ８０以降の処理を繰り返す。

【０１０１】すべての部分線分矩形について処理が終われば、ｂ＿ｕｓｅ＝９であるかどうかを判定し（ステップＳ１００）、ｂ＿ｕｓｅ＝９であれば、ｈｅｉｇｈｔ＝ｈｅｉｇｈｔ２とおく（ステップＳ１０１）。ステップＳ１００でｂ＿ｕｓｅ＝９となるのは、カレント矩形ｉとそれに連結するすべての矩形がワイルドカードであ

った場合に相当する。

【０１０２】次に、カレント矩形ｉとしてすべての部分線分矩形を取り出したかどうかを判定し（ステップＳ１０２）、残っている部分線分矩形があればステップＳ７５以降の処理を繰り返す。ステップＳ７６でカレント矩形ｉのｕｓｅが０または９でない場合は、取り出した部分線分矩形が既に使用済みであることを意味するので、直ちにステップＳ１０２の処理に移り、次の部分線分矩形を取り出す。

【０１０３】すべての部分線分矩形について処理が終われば、ｘｌｆ、ｘｒ、ｌｉｎｅ＿ｓｔａｒｔｙ、ｌｉｎｅ＿ｅｎｄｙを、それぞれ抽出した線分矩形の左端、右端、上端、下端の座標としてファイルyokolineに保存し（ステップＳ１０３）、処理を終了する。ここで、yokolineは、１つの文字列矩形から抽出された１つ以上の線分矩形を格納するメモリ領域に対応する。

【０１０４】図２４のステップＳ５０では、以上のようにして文字列矩形から線分が抽出され、さらにそれが下線矩形に相当すれば、その文字列矩形に下線フラグが立てられる。こうして文字列矩形加工処理が終了すると、プロセッサ１４は、次に図３のステップＳ８～Ｓ１０の処理に相当するタイトル・宛先・発信元抽出処理を行う。図４０は、タイトル・宛先・発信元抽出処理のフローチャートである。

【０１０５】図４０において処理が開始されると、まず文字列矩形の相対的な位置、高さ、枠／下線情報を使って、タイトルらしさのポイント計算を行う（ステップＳ１１１）。各文字列矩形に対するタイトルらしさのポイント付与の方針は、概ね次の通りである。

（ｍ）プラスポイント

文字列の属性（枠内、下線有り）：高得点

文字列のサイズ（高さ、幅）：大きさに依存する得点

文字列の形（縦横比）：一定以上であれば得点

文字列の相互位置関係（上下間隔、左の矩形の有無）：孤立性が高いほど高得点

文書内の位置（中央、上など）：中央、上は高得点、上下の位置の違いには相対的に少ない得点差

（ｎ）マイナスポイント

文字列の属性（文字列矩形内が１つの文字矩形からなる）：大減点

文字列の相互位置関係（上下近接、重複、上の矩形と左揃い、上の矩形がオーバラップ）：大減点

文書内の位置（右側にある）：大減点

これらの方針に従い、各文字列矩形に例えば以下の条件で得点を与える。

（ｏ）罫線矩形は得点０

（ｐ）高さが str__freqの１／２未満は得点０

（ｑ）幅／高さの比が３未満は得点０

（ｒ）横幅が str__freqの４倍未満は得点０

（ｓ）（ｏ）、（ｐ）、（ｑ）、（ｒ）の条件に該当する文字列矩形以外のものについて、以下の条件で得点を与える。

【０１０６】［＃１］縦横比：幅／高さの比が３の時、２０点

［＃２］上下近接：互いに重複している場合を除き、ある文字列矩形と、上下に隣接する２つの文字列矩形との間隔が両方とも str__freq／２以下の時、－４０点

［＃３］片方近接：上または下の文字列矩形だけが１６ドットより近接している場合、－２０点

［＃４］上下間隔：上下の文字列矩形との間隔が str__freqより大きい場合、２０点

［＃５］重複：他の文字列矩形と重複がある場合、－４０点

［＃６］中心：文字列矩形の横方向（ｘ方向）の中心座標が、（文書領域の中心座標）±（文書領域幅の４０％）以内に入っている場合、３０点

［＃７］右側：文字列矩形の中心座標が、文書領域の左から６０％の位置より右にあり、かつ、（文書領域の中心座標－文字列矩形の左端座標）が文書領域幅の１／６以下の場合、３０点

［＃８］高さ１：文字列矩形の高さが str__freqの０．５倍から１．５倍の間にある場合、２０点

［＃９］高さ２：文字列矩形の高さが str__freqの１．５倍と３倍の間の場合３０点

［＃１０］高さ３：文字列矩形の高さが str__freqの３倍より大きい場合、４０点

［＃１１］高さ４：文字列矩形の高さが str__freqの３倍より大きく、かつ、文字列矩形の下座標が文書領域の上から１／３以内に入る場合、１０点

［＃１２］横幅：文字列矩形の幅が文書領域幅の０．４倍より大きい場合、１０点

［＃１３］下線：文字列矩形に下線フラグがある場合、３０点

［＃１４］枠：文字列矩形に枠付きフラグがある場合、最大３０点を与え、その横幅に比例して減少させる。

【０１０７】［＃１５］左に矩形が無い：左側に同じような座標の文字列矩形が無い場合、または、左側に str__freqの３倍より小さい文字列矩形がある場合、２０点

［＃１６］ｙ座標：最も上にある文字列矩形が２０点、そこから下に向かって１点づつ減少した得点

［＃１７］左端揃い：文字列矩形の上に左端が近い他の文字列矩形があると－３０点

［＃１８］オーバラップ（overlap）：文字列矩形の上に左端および右端が近い他の文字列矩形がある場合、または、上の文字列矩形の方が左端右端とも文書領域の端に近い場合、－３０点

［＃１９］黒領域：大きな文字列矩形で、その内部が１つの黒画素連結領域で成り立っている場合、－４０点

図４１は、［＃１８］のオーバラップしている文字列矩形の例を示している。図４１（ａ）においては、上の文

字列矩形と下の文字列矩形の左右端が近接しており図４１（ｂ）においては、上の文字列矩形の左右端の方が、下の文字列矩形の左右端より文書領域の端に近い。このような場合、下の文字列矩形はタイトルである可能性が低いと考えられる。

【０１０８】上記（ｏ）、（ｐ）、（ｑ）、（ｒ）、（ｓ）のポイントを、文字列矩形毎に合計し、メモリ１５に保存する。次に、ポイントが高い順にタイトル候補として抽出し、結果を保存する（ステップＳ１１２）。ここでは、line3 内の全文字列矩形を対象にして、それらをポイントが高い順に並び替え、その結果をファイル title に格納する。title 内には、タイトル候補の第１位の文字列矩形から順に、全文字列矩形が格納される。これにより、第１候補の文字列矩形がタイトル矩形として抽出される。次に、タイトル候補の第１位の文字列矩形から見た相対的位置関係の情報を使って宛先の文字列矩形（宛先矩形）を抽出し、保存する（ステップＳ１１３）。また、その相対的位置関係の情報または宛先矩形から見た相対的位置関係の情報を使って発信元情報の文字列矩形（発信元情報矩形）を抽出し、それを保存して（ステップＳ１１４）、処理を終了する。発信元情報には、文書の発信日、発信者名、レポート番号等が含まれる。

【０１０９】ステップＳ１１３においては、まずタイトルの第１候補の文字列矩形のｙ方向の位置を求めて、それが最も上であった場合は第１の宛先抽出処理を行い、それ以外の場合は第２の宛先抽出処理を行う。図４２は、第１の宛先抽出処理のフローチャートであり図４３は、第２の宛先抽出処理のフローチャートである。

【０１１０】まず、第１の宛先抽出処理について説明する。図４２において処理が開始されると、プロセッサ１４は、まずタイトル矩形より下にある文字列矩形の中からキー宛先矩形を抽出し、それを保存する（ステップＳ１２１）。ここでは、タイトル矩形より下にあり、高さがst_h の０．６倍からen_h の１．４倍の間にある文字列矩形であって、そのｘ方向の中心座標がタイトル矩形の中心座標よりも左にあり、幅／高さの比が３より大きいものを、キー宛先矩形として抽出する。そして、このキー宛先矩形より上にある文字列矩形の中に、ｘ方向の中心座標がタイトル矩形の中心座標よりも右にあるような、発信元情報と思える文字列矩形がない場合に、抽出したキー宛先矩形をファイルtoに保存する。

【０１１１】次に、キー宛先矩形の右にある文字列矩形を宛先矩形として追加する（ステップＳ１２２）。ここでは、キー宛先矩形の右にあり、そのｙ座標が（キー宛先矩形のｙ座標）±（高さの０．２倍）の範囲内に収まっている文字列矩形を宛先矩形とみなし、キー宛先矩形との重複登録を避けて、toに登録する。

【０１１２】次に、上下に宛先矩形がある文字列矩形を宛先矩形として追加する（ステップＳ１２３）。ここで

は、これまで抽出したto内の宛先矩形の高さの平均値（平均高さ）を求める。そして、タイトル矩形より下の全文字列矩形の内、これまで抽出された宛先矩形でなく、上または下が宛先矩形で、左端の座標が上または下の宛先矩形の左端の座標と一定誤差以内で一致し、かつ、高さが平均高さの２倍未満か、上または下の宛先矩形までの距離が平均高さの１／２未満のものを、宛先矩形としてtoに追加登録する。このような処理を宛先矩形数が変化しなくなるまで繰り返す。

【０１１３】こうして、第１の宛先抽出処理が終了し、to内の文字列矩形が宛先矩形として抽出される。次に、第２の宛先抽出処理について説明する。図４３において処理が開始されると、プロセッサ１４は、まずタイトル矩形より上にある文字列矩形の中からキー宛先矩形を抽出し、それを保存する（ステップＳ１３１）。ここでは、タイトル矩形より上にあり、高さがst_h の０．６倍からen_h の１．４倍の間にある文字列矩形であって、そのｘ方向の中心座標がタイトル矩形の中心座標よりも左にあり、幅／高さの比が３より大きいものを、キー宛先矩形として抽出する。そして、抽出したキー宛先矩形をファイルtoに保存する。

【０１１４】次に、キー宛先矩形の右にある文字列矩形を宛先矩形として追加する（ステップＳ１３２）。ここでは、キー宛先矩形の右一定距離以内にあり、そのｙ座標が（キー宛先矩形のｙ座標）±（高さの０．２倍）の範囲内に収まっている文字列矩形を宛先矩形とみなし、キー宛先矩形との重複登録を避けて、toに登録する。

【０１１５】次に、上下に宛先矩形がある文字列矩形を宛先矩形として追加する（ステップＳ１３３）。ここでは、これまで抽出したto内の宛先矩形の平均高さを求める。そして、タイトル矩形より下の全文字列矩形の内、これまで抽出された宛先矩形でなく、上または下が宛先矩形で、左端の座標が上または下の宛先矩形の左端の座標と一定誤差以内で一致し、かつ、高さが平均高さの２倍未満か、上または下の宛先矩形までの距離が平均高さの１／２未満のものを、宛先矩形としてtoに追加登録する。このような処理を宛先矩形数が変化しなくなるまで繰り返す。

【０１１６】こうして、第２の宛先抽出処理が終了し、to内の文字列矩形が宛先矩形として抽出される。図４０のステップＳ１１４では、タイトル矩形のｙ方向の位置を求めて、それが最も上であった場合は、第１の発信元情報抽出処理を行い、それ以外の場合は第２の発信元情報抽出処理を行う。

【０１１７】第１の発信元情報抽出処理においては、タイトル矩形より下の文字列矩形であって、宛先矩形でないものを対象にして、高さがst_h の０．６倍からen_h の１．４倍の間にあり、かつ、ｘ方向の中心座標がタイトル矩形のそれよりも右にあるものを、宛先矩形として抽出し、ファイルfromに保存する。また、第２の発信

元情報抽出処理においては、タイトル矩形より上の文字列矩形であって、宛先矩形でないものを対象にして、第１の発信元情報抽出処理と同様の文字列矩形を宛先矩形として抽出し、ファイルfromに保存する。こうして、from内の文字列矩形が発信元情報矩形として抽出される。

【０１１８】第１および第２の発信元情報抽出処理は、第１および第２の宛先抽出処理に比べて簡単になっているが、宛先抽出処理と同様に、一定の条件を満たす他の文字列矩形をさらに発信元情報矩形に加えるようにしてもよい。

【０１１９】図４４は、タイトルと宛先／発信元情報の第１の配置を示している。図４４においては、タイトル矩形が最も上にあるため、第１の宛先抽出処理および第１の発信元情報抽出処理が適用される。図４５、４６、４７は、それぞれタイトルと宛先／発信元情報の第２、第３、第４の配置を示している。これらの配置においては、タイトル矩形が最も上ではないので、第２の宛先抽出処理および第２の発信元情報抽出処理が適用される。また、図４８は、複数の宛先／発信元情報の例を示している。図４８においても、第２の宛先抽出処理および第２の発信元情報抽出処理が適用される。

【０１２０】図４５、４７、４８のような配置の場合は、第２の発信元情報抽出処理を行うと、タイトル矩形より下にある発信元情報矩形が抽出されない。そこで、タイトル矩形が最も上にない場合でも、第１の発信元情報抽出処理を行う構成としてもよい。また、第１および第２の発信元情報抽出処理を併用してもよい。

【０１２１】図４９は、タイトル・宛先・発信元抽出処理により生成されたファイルtitle、to、fromの内容を示している。図４９においては、文字列矩形「ソフトウェア販推レポート　送付表」がタイトル矩形として抽出され、それに続く左揃いの文字列矩形が複数の宛先矩形として抽出されている。また、右下の数字が発信元情報として抽出されている。

【０１２２】図５０は、タイトル・宛先・発信元抽出処理による他の抽出結果を示している。図５０においては、文字列矩形「外部発表の受付状況について（送付）」がタイトル矩形として抽出され、その左上にある文字列矩形が宛先矩形として抽出されている。また、タイトル矩形の右上の複数の文字列矩形が発信元情報として抽出されている。

【０１２３】こうして、抽出されたタイトル矩形、宛先矩形、および発信元情報矩形は、図３のステップＳ１１の認識処理により文字列として認識される。このとき、各認識対象の矩形から１文字ずつ文字が切り出され、各文字毎に文字認識が行われる。そして、認識結果は、例えば、電子ファイリング装置１１内の画像ファイルのキーワードとして用いられる。

【０１２４】以上の実施形態において、図３１の線分抽出処理は、図２４のステップＳ５０の下線抽出処理のみならず、図３のステップＳ６で大きな矩形から横線分を抽出する際にも適用できる。これにより、大きな矩形内のワイルドカードの高さを無視して横方向の線分矩形を抽出し、それを一部分とする枠線を識別することができる。

【０１２５】ところで、図３から図５０までで説明した実施形態では、表の外部にある領域からタイトルを抽出する技術について記述されている。表の内部にタイトルがある場合には、図５のステップＳ２４で表矩形が処理対象から除外されているため、表内のタイトルを抽出することができない。

【０１２６】一般に表を含む文書においては、その表の外に文書全体のタイトルがあることが多いが、会社内の文書の中には、定型の事務文書等のように表の内部にタイトルがあるものもある。また、表の外にタイトルがあっても、それが「議事録」などのような一般的な文書名で、電子ファイリングシステムの検索時に必要な文書を特定できるキーワードとなるタイトルは表内の１つの欄内に記されていることもある。

【０１２７】このような場合に、文字認識などの処理時間のかかる技術を使わずに、表内の有効なタイトル部分を高速に抽出することが望まれる。以下では、表を含む一般文書の文書画像から、表内にある「表題」や「会社名」のようなタイトルらしい欄の名称を表現する部分（項目部分）と、項目の具体的な内容を表すタイトル部分とを抽出する実施形態を説明する。

【０１２８】図５１は、表形式の社内文書の例を示している。図５１の表形式文書では、表罫線で囲まれた表内の左上にある「表題」が項目部分に相当し、その右にある「マルチメディアとパターン認識シンポジウム」がタイトル部分に相当する。このように、横書き文書の場合には、表内のタイトル部分は、通常、項目部分の右側にあると考えてよい。

【０１２９】図５２は、図２のタイトル抽出システムによる表内タイトル抽出処理のフローチャートを示している。図５２の処理においては、前提条件として横書き文書を対象としているが、図３の処理と同様に、縦書き文書にも対応可能である。

【０１３０】図５２において処理が開始されると、光電変換装置１２が文書を読み取り、文書画像としてメモリ１５に格納する（ステップＳ１４１）。ここでも図３のステップＳ１と同様にして、原画像を圧縮画像に変換して保存する。図５１の文書から作成された圧縮画像は図５３のようになる。

【０１３１】次に、プロセッサ１４が、文書画像にラベリング処理を施し、矩形高さの最頻値を求めて、それをもとに大きな矩形を抽出する（ステップＳ１４２）。ここでの処理は、図５のステップＳ２１、Ｓ２２、Ｓ２３の処理と同様である。ただし、枠矩形の抽出は行っておらず、ファイルbox に保存される矩形は閾値th_large

より大きな矩形である。図５３の文書画像のラベリング結果は図５４のようになる。

【０１３２】次に、抽出された大きな矩形から表を囲む矩形（表矩形）を抽出し（ステップＳ１４３）、表矩形の中からタイトルを含むものを選択する（ステップＳ１４４）。ここでは、例えば最も面積の大きな表矩形が選択され、以下の処理は選択された表矩形の内部を対象にして行われる。

【０１３３】プロセッサ１４は、まず表矩形の内部から文字列（あるいは行）を抽出し、文字列の外接矩形（文字列矩形）を求め、その座標をメモリ１５に保存する（ステップＳ１４５）。次に、保存した文字列矩形から、横幅が小さい矩形や縦長矩形をノイズ矩形として除去し（ステップＳ１４６）、２つ以上の文字列矩形を統合する（ステップＳ１４７）。

【０１３４】ステップＳ１４５の処理は、基本的に図５のステップＳ２５からＳ３１までの処理と同様である。また、ステップＳ１４６の処理は、図２４のステップＳ４１の処理と同様であり、ステップＳ１４７の処理は、ステップＳ４２からＳ４４までの処理と同様である。

【０１３５】ここまでの処理で、表内から抽出された文字列矩形が整理されるが、これらの文字列矩形は表罫線の一部を含んでいる場合もあり得る。そこで、文字列矩形の中の罫線部分を抽出し、その部分を境にして文字列矩形を分割する（ステップＳ１４８）。

【０１３６】次に、タイトルに相当する文字列矩形を抽出するために、文字列矩形内の文字数を計算する（ステップＳ１４９）。ここで計算された文字数は、文字列矩形の属性としてステップＳ１５２の処理で用いられる。

【０１３７】ステップＳ１４８の処理により表罫線で囲まれた欄毎の文字列矩形が抽出されるが、元の表の外形が矩形ではない場合には、表の外にある文字列矩形が残されている可能性がある。そこで、上罫線のチェックを行って（ステップＳ１５０）、上側に表罫線がないような文字列矩形は表外の文字列矩形とみなし、それを除去する。

【０１３８】次に、表内の文字列矩形を表矩形の左上座標に近い順に並び替える（ステップＳ１５１）。そして、文字列矩形の文字数が一定の条件を満たす場合に、その文字列矩形を項目部分またはタイトル部分として抽出して（ステップＳ１５２）、処理を終了する。このとき、条件を満たす文字列矩形を、表矩形の左上に近いものから優先的にタイトル矩形の候補とする。

【０１３９】次に、表内タイトル抽出処理の各ステップで行われる具体的な処理内容を説明する。図５５は、図５２のステップＳ１４３の表矩形抽出処理のフローチャートである。この表矩形抽出処理に先立ってステップＳ１４２の処理を行っておくことで、処理対象が一定以上大きな矩形に限られるため、表矩形の抽出が効率化される。

【０１４０】図５５において処理が開始されると、プロセッサ１４は、まずbox 内の大きな矩形から高さが閾値より大きなものを抽出する（ステップＳ１６１）。ここでは、例えば矩形高さの最頻値freqの５倍より大きい（高い）矩形が抽出され、表矩形としてファイル large__4baiに格納される。ステップＳ１６１で抽出された表矩形は、ステップＳ１５０の上罫線チェックの際に用いられる。

【０１４１】次に、box 内の大きな矩形から横幅が閾値より大きなものを抽出して（ステップＳ１６２）、処理を終了する。ここでは、例えば横幅が文書画像の横幅の０．７倍より大きな矩形が抽出され、表矩形としてファイルlargewide に格納される。

【０１４２】図５２のステップＳ１４４では、ステップＳ１６２で抽出されたいくつかの表矩形のうちで最も大きなものが選択される。ここでは、例えば、largewide 内の複数の矩形からその面積が最大のものが選択されて、処理対象となる。図５４の文書画像の場合は、largewide 内に格納される大きな矩形は表矩形８０のみであるため、自動的にこれが処理対象の表矩形となる。

【０１４３】次に、図５２のステップＳ１４５では、選択された表矩形内部の文字矩形を対象にして文字列矩形の抽出が行われる。しかし、次のいずれかの条件に該当する矩形は処理対象から除外する。

（ｔ）枠矩形

（ｕ）高さがfreqの３倍より大きく、高さ／幅の比が０．４より小さな横長矩形（ｖ）文書画像全体の高さの１／３より大きな矩形

このうち、（ｔ）の枠矩形は、図５のステップＳ２３と同様の処理により抽出することができる。

【０１４４】ステップＳ１４５、Ｓ１４６、Ｓ１４７の処理を行った後に得られる統合された文字列矩形は図５６のようになる。図５６において、例えば文字列矩形８１、８２、８３等は、表罫線により仕切られた本来別々の複数の文字列を含んでいる。そこで、表内の文字列を正しく抽出するために、ステップＳ１４８で文字矩形間の縦罫線を境界にして文字列矩形を分割する。以下、図５７から図６５までを参照しながら、この文字列分割処理について説明する。

【０１４５】文字列分割方法としては、大きく分けて２つの方法が考えられる。図５７は、第１の文字列分割処理のフローチャートである。第１の文字列分割処理においては、プロセッサ１４は、各文字列矩形に含まれる任意の２つの隣接文字矩形の間に縦罫線があるかどうかをチェックする。このとき、まず文字列矩形内に含まれる文字矩形を横方向に並び替え、それらの間に黒画素があるかどうかチェックする。黒画素がある場合はその位置で文字列矩形を分割して、複数の新しい文字列矩形を生成する。

【０１４６】図５７において処理が開始されると、プロ

セッサ１４は、まず文字列矩形内の文字矩形をｘ座標（横座標）の小さいものから順にソートする（ステップＳ１７１）。ステップＳ１４７までの処理においては、文字列矩形内の文字矩形は一般にｙ座標（縦座標）の小さい順にソートされており、横方向の順序が反映されていない。そこで、実際の文字の並びに対応するように、文字矩形の記憶順序が変更される。

【０１４７】例えば、図５８に示す文字列矩形９１の場合、文字列分割処理の前には文字矩形９２、９５、９３、９４の順にソートされて、記憶されている。これら10の文字矩形をｘ座標でソートし直すことにより図５９に示すように文字矩形９２、９３、９４、９５の順に正しく記憶される。

【０１４８】次に、文字列矩形の左端のｘ座標、右端のｘ座標、上端のｙ座標、下端のｙ座標を、それぞれｓｘ１、ｓｘ２、ｓｙ１、ｓｙ２とおき（ステップＳ１７２）、文字列矩形内の最も左の文字矩形に注目し、それをカレント矩形とする（ステップＳ１７３）。そして、カレント矩形の上端のｙ座標、下端のｙ座標、右端のｘ座標を、それぞれｃｙ１、ｃｙ２、ｃｘ２とおき（ステ20ップＳ１７４）、カレント矩形の右にある文字矩形の上端のｙ座標、下端のｙ座標、左端のｘ座標を、それぞれｒｙ１、ｒｙ２、ｒｘ１とおく（ステップＳ１７５）。

【０１４９】次に、直線ｘ＝ｃｘ２，ｘ＝ｒｘ１，ｙ＝ｍａｘ（ｃｙ１，ｒｙ１），ｙ＝ｍｉｎ（ｃｙ２，ｒｙ２）で囲まれた矩形領域内に黒画素があるかどうかをチェックする（ステップＳ１７６）。ここで、この矩形領域は、カレント矩形とカレント矩形の右の文字矩形の間に位置する領域である。

【０１５０】上記矩形領域内に黒画素があれば、そこに 30縦罫線があるとみなして、座標ｘ＝ｓｘ１，ｃｘ２，ｙ＝ｓｙ１，ｓｙ２で表される矩形を文字列矩形として登録し、ｓｘ１＝ｒｘ１とする（ステップＳ１７７）。

【０１５１】次に、カレント矩形の右の文字矩形が文字列矩形の中で最も右にあるかどうかを調べ（ステップＳ１７８）、そうでない場合はカレント矩形の右の文字矩形を新たにカレント矩形として（ステップＳ１７９）、ステップＳ１７４以降の処理を繰り返す。ステップＳ１７６で上記矩形領域内に黒画素がなければ、そのままステップＳ１７８以降の処理を行う。 40

【０１５２】そして、ステップＳ１７８においてカレント矩形の右の文字矩形が最も右の矩形である場合は、座標ｘ＝ｓｘ１，ｓｘ２，ｙ＝ｓｙ１，ｓｙ２で表される矩形を文字列矩形として登録して（ステップＳ１８０）、処理を終了する。

【０１５３】このような第１の文字列分割処理によれば、カレント矩形とカレント矩形の右の矩形の間に縦罫線が検出される度に、その左側の１つ以上の文字矩形が文字列矩形として登録される。したがって、元の文字列矩形に縦罫線が２本以上含まれていても、必ずそれらの50

位置で文字列矩形が分割される。

【０１５４】例えば、図６０のような表内の文字列矩形１０１の場合、文字矩形１０２、１０３、１０４、１０５、１０６、１０７を含んでおり、文字矩形１０２と文字矩形１０３の間には表の縦罫線が通っている。この文字列矩形１０１を対象に第１の文字列分割処理を行うと、文字矩形１０２がカレント矩形のとき、文字矩形１０２と文字矩形１０３の間の領域に黒画素が検出される（ステップＳ１７６、Ｙｅｓ）。そこで図６１に示すように、文字矩形１０２を含む矩形が文字列矩形１０８として登録される（ステップＳ１７７）。

【０１５５】その後、文字矩形１０３が新たにカレント矩形となって（ステップＳ１７９）、同様の処理が繰り返されるが、縦罫線は検出されない。そして、文字矩形１０６がカレント矩形となったとき、文字矩形１０３、１０４、１０５、１０６、１０７を含む矩形が文字列矩形１０９として登録され（ステップＳ１８０）、処理が終了する。こうして、元の文字列矩形１０１は、文字列矩形１０８と１０９に分割される。

【０１５６】図６２および図６３は、第２の文字列分割処理のフローチャートである。第２の文字列分割処理においては、プロセッサ１４は、各文字列矩形の内部を対象にして再度ラベリング処理を施す。このとき、まず文字列矩形を構成する各文字矩形の座標を記憶しておき、それとは別に、文字列矩形内のラベリング処理により得られた文字矩形の座標を獲得する。

【０１５７】縦罫線の一部が文字列矩形内にあるとすると、前者の文字矩形群と後者の文字矩形群とを比較した場合、後者の方が縦罫線の分だけ矩形の数が増えるため、両者の間に差異が生じるはずである。そこで、前者と比較して後者に余分な文字矩形が出現した位置で文字列矩形を分割する。

【０１５８】例えば、図６０の文字列矩形１０１の場合、その内部にラベリング処理を施して得られる文字矩形は図６４のようになる。図６０の文字矩形群と図６４の文字矩形群とを比較すると、図６４の方が余分な矩形１１０を含んでいることが分かる。この矩形１１０は文字列矩形１０１内に含まれた縦罫線に相当し、この位置で文字列矩形１０１を分割することができることを表している。

【０１５９】図６２において処理が開始されると、プロセッサ１４は、まず文字列矩形内の文字矩形の集合をＯとし（ステップＳ１８１）、文字列矩形内をラベリング処理して求めた文字矩形の集合をＮとする（ステップＳ１８２）。そして、集合ＯとＮ内の文字矩形をそれぞれｘ座標でソートし（ステップＳ１８３）、文字列矩形の左端のｘ座標、右端のｘ座標、上端のｙ座標、下端のｙ座標を、それぞれｓｘ１、ｓｘ２、ｓｙ１、ｓｙ２とおく（ステップＳ１８４）。ｘ座標によるソート処理は、図５７のステップＳ１７２と同様にして行う。

35

【０１６０】次に、登録フラグ＝０とおき、O内の最も左の文字矩形を矩形ＯＯとし、N内の最も左の文字矩形を矩形ＮＮとする。そして、ｘ２＝ＯＯの右端のｘ座標、ｐｒｅｖ＝ｘ２とおく（ステップＳ１８５）。以後、登録フラグは０または１の値をとる。

【０１６１】次に、ＯＯとＮＮの左上頂点および右下頂点の座標が一致するかどうかをチェックする（ステップＳ１８６）。これらがともに一致すればＯＯとＮＮは同じ矩形であるとみなし、次に、登録フラグが１かどうかを判定する（ステップＳ１８７）。

【０１６２】登録フラグが０の場合は、ＯＯの右の矩形を新たにＯＯとおき、ＮＮの右の矩形を新たにＮＮとおく（ステップＳ１８８）。そして、ｐｒｅｖ＝ｘ２とおいた後（ステップＳ１８９）、ｘ２＝ＯＯの右端のｘ座標とおき（ステップＳ１９０）、ＯＯが文字列矩形の中で最も右の文字矩形かどうかを判定する（ステップＳ１９１）。そして、ＯＯの右にまだ文字矩形があれば、ステップＳ１８６以降の処理を繰り返す。

【０１６３】ステップＳ１８６において、ＯＯとＮＮの座標が一致しない場合はＮＮが縦罫線に相当するとみなし、次に登録フラグが０かどうかを判定する図６３、ステップＳ１９５）。そして、登録フラグが０であれば、座標ｘ＝ｓｘ１，ｐｒｅｖ，ｙ＝ｓｙ１，ｓｙ２で表される矩形を文字列矩形として登録し（ステップＳ１９６）、登録フラグ＝１とおく（ステップＳ１９７）。これにより、ＯＯの左の文字矩形を含む矩形が文字列矩形として登録される。

【０１６４】次に、縦罫線とみなされたＮＮの右の矩形を新たにＮＮとおいて（ステップＳ１９８）、ステップＳ１８６以降の処理を繰り返す。ステップＳ１９５において登録フラグが０でなければ、そのままステップＳ１９８以降の処理を行う。

【０１６５】ステップＳ１８７において、登録フラグが１の場合はＯＯを新たな文字列の先頭文字とみなして、ｘ２＝ＯＯの右端のｘ座標、ｓｘ１＝ＯＯの左端のｘ座標とおく（ステップＳ１９２）。そして、ｐｒｅｖ＝ｘ２、登録フラグ＝０とおき（ステップＳ１９３、Ｓ１９４）、ステップＳ１９１以降の処理を行う。

【０１６６】そして、ステップＳ１９１においてＯＯが最も右の文字矩形の場合は、座標ｘ＝ｓｘ１，ｘ２，ｙ＝ｓｙ１，ｓｙ２で表される矩形を文字列矩形として登録して（ステップＳ１９９）、処理を終了する。

【０１６７】このような第２の文字列分割処理によれば、集合N内にあってO内にはない余分な矩形が検出される度に、その左側の１つ以上の文字矩形が文字列矩形として登録される。また、その後はO内の次の矩形が文字列の左端に設定されるので、余分な縦罫線は文字列矩形から除去される。

【０１６８】例えば、図６４の文字列矩形１０１の場合、集合Oは文字矩形１０２、１０３、１０４、１０５、１０６、１０７からなり、集合Nは文字矩形１０２、１１０、１０３、１０４、１０５、１０６、１０７からなる。そして、ＯＯが文字矩形１０３でＮＮが文字矩形１１０のとき、文字矩形１１０が縦罫線とみなされる（ステップＳ１８６、Ｎｏ）。そこで図６１に示すように、文字矩形１０２を含む矩形が文字列矩形１０８として登録される（ステップＳ１９６）。

36

【０１６９】その後、文字矩形１０３が新たにＮＮとなって（ステップＳ１９８）、同様の処理が繰り返されるが、縦罫線に相当する矩形は検出されない。そして、文字矩形１０７がＯＯとなったとき、文字矩形１０３、１０４、１０５、１０６、１０７を含む矩形が文字列矩形１０９として登録され（ステップＳ１９９）、処理が終了する。こうして、元の文字列矩形１０１は、第１の文字列分割処理の結果と同様に、文字列矩形１０８と１０９に分割される。

【０１７０】第１および第２の文字列分割処理を比較すると、それらの機能は基本的に同じであるが、第１の文字列分割処理の方が処理速度が速いという利点がある。図５６の文字列矩形に文字列分割処理を施した結果図６５のようになる。図５６と図６５とを比較すると、元の文字列矩形８１は文字列矩形１１１、１１２、および１１３に分割されていることが分かる。また、文字列矩形８２は文字列矩形１１４と１１５に分割され、文字列矩形８３は文字列矩形１１６と１１７に分割されている。

【０１７１】文字列矩形の分割が終了すると、次に、図５２のステップＳ１４９において、プロセッサ１４は、文字列矩形内の文字矩形の形状からその文字数を計算する。ここでは、文字矩形の高さと幅の比からそれを構成する文字数を抽出する。

【０１７２】図６６は、このときの文字矩形とその文字数の関係を示している。図６６において、文字矩形の高さをH、幅をWとすると、一般に１つの文字の高さと幅はほぼ等しいと考えられるので、この文字矩形内にある文字数は［W／H］個と表すことができる。ここで、［W／H］は、実数W／Hの小数点以下を切り捨てる演算記号である。

【０１７３】ステップＳ１４８の文字列分割処理により表矩形内の文字列矩形が正しく分割されるが、表矩形内には実際の表の外にある文字列矩形が含まれている可能性がある。図６７は、このような表矩形内の表外文字列矩形の例を示している。図６７において、太線で示された表罫線の外周は矩形ではないため、その表矩形１２１内には表外にある文字列矩形１２２が含まれている。一方、文字列矩形１２２と同じ行にある文字列矩形１２３は表内の文字列矩形である。

【０１７４】図６８は、図５４の表矩形８０内の文字列矩形を示している。図６８の文字列矩形のうち、文字列矩形１３１が表外の文字列矩形に相当する。表内のタイ

トルを抽出するためには、文字列矩形１２２や１３１のような表外の文字列矩形を表内の文字列矩形と区別し、表矩形内から取り除く必要がある。

【０１７５】そこで、ステップＳ１５０において、上に他の文字列矩形がない文字列矩形を対象に、その上に罫線があるかどうかをチェックし、罫線がなければその文字列矩形を除去する。

【０１７６】図６９は、このような上罫線チェック処理のフローチャートである。図６９において処理が開始されると、プロセッサ１４は、ま図２４のステップＳ４２と同様の方法で、文字列矩形間の接続関係を表す連結関係表を作成する（ステップＳ２０１）。そして、連結関係表を用いて上に他の文字列矩形がない文字列矩形を求め、それらのうちで上に罫線がないものを除去して（ステップＳ２０２）、処理を終了する。

【０１７７】図７０は、ステップＳ２０２の表外文字列矩形除去処理のフローチャートである。図７０の表外文字列矩形除去処理においては、表矩形内のすべての文字列矩形の連結関係表を参照して、文字列矩形の上に他の文字列矩形がないものを抽出する。そして、抽出した文字列矩形の上の特定領域内を探索して、黒画素を含むバイト数の合計Ｍを求める。ただし、８画素＝１バイトとする。

【０１７８】Ｍが探索範囲の横の長さをバイト数で表したしきい値Ｌ以上であれば、この範囲に横罫線があるとみなして、その文字列矩形を表内文字列矩形として残す。もし、Ｍ＜Ｌとなるような文字列矩形があれば、その文字列矩形の上には横罫線がないとみなし、それを表外文字列矩形として除去する。

【０１７９】図７０において処理が開始されると、プロセッサ１４は、まず表矩形内の文字列矩形からなる集合を、表内文字列矩形の集合Ｓとする（ステップＳ２１１）。次に、Ｓ内で、他のＳ内の文字列矩形を上矩形とする接続関係を持たないものを抽出し、それらの集合をＳ１とする（ステップＳ２１２）。例えば図６７の場合は、斜線の文字列矩形１２２と１２３がＳ１の要素となる。

【０１８０】次に、Ｓ１内の１つの文字列矩形をＳＳとし（ステップＳ２１３）、ＳＳの左端のｘ座標、右端のｘ座標、上端のｙ座標、下端のｙ座標を、それぞれｓｘ１、ｓｘ２、ｓｙ１、ｓｙ２とおく（ステップＳ２１４）。

【０１８１】次に、ＳＳの上にある他の表矩形または表矩形外の文字列矩形を求め、その左端のｘ座標、右端のｘ座標、上端のｙ座標、下端のｙ座標を、それぞれｕｘ１、ｕｘ２、ｕｙ１、ｕｙ２とおく（ステップＳ２１５）。ここで、他の表矩形としては、図５５のステップＳ１６１で抽出されて large__4baiに格納されている表矩形が参照される。

【０１８２】次に、直線ｘ＝ｍａｘ（ｓｘ１，ｕｘ１），ｘ＝ｍｉｎ（ｓｘ２，ｕｘ２），ｙ＝ｓｙ１，ｙ＝ｕｙ２で囲まれた矩形領域の横幅のバイト数をＬとする（ステップＳ２１６）。この矩形領域の横幅は、ＳＳの横幅とその上の矩形の横幅の重複部分の長さに相当する。このとき、Ｌは次式で与えられる。

Ｌ＝ｍｉｎ（ｓｘ２，ｕｘ２）／８－ｍａｘ（ｓｘ１，ｕｘ１）／８＋１

次に、上記矩形領域内で黒画素を求め、８画素を１バイトとして、そのバイト数の総和Ｍを計算する（ステップＳ２１７）。

【０１８３】ステップＳ２１５で求めたＳＳの上にある矩形が文字列矩形の場合は、ステップＳ２１６のＬと、ステップＳ２１７における黒画素の探索範囲は図７１のようになる。また、ＳＳの上にある矩形が他の表矩形の場合は、それらに図７２のようになる。

【０１８４】次に、ＭとＬの大きさを比較し（ステップＳ２１８）、ＭがＬ未満であればＳＳの上に横罫線がないものとみなして、ＳＳを表外の文字列矩形と判定する。そこで、ＳＳを集合Ｓから除去する（ステップＳ２１９）。

【０１８５】次に、ＳＳがＳ１内の最後の文字列矩形かどうかを判定し（ステップＳ２２０）、最後でなければステップＳ２１３以降の処理を繰り返す。そして、Ｓ１内の文字列矩形をすべて処理すると、処理を終了する。

【０１８６】ステップＳ２１５において、ＳＳの上に表矩形も文字列矩形もない場合は、ステップＳ２１７で文書画像の上端までの範囲を探索して、黒画素を求めればよい。このときの探索範囲は図７３のようになり、その横幅はＳＳの横幅に一致する。

【０１８７】図７０の処理により、図６８の表外文字列矩形１３１が除去され、残された表内文字列矩形図７４のようになる。こうして得られた表内文字列矩形を対象にして、それらの位置や文字数の関係からタイトルの候補が抽出される。

【０１８８】図７５は、図５２のステップＳ１５１およびＳ１５２で行われるタイトル候補出力処理のフローチャートである。横書き文書の場合は、一般に左上に近い文字列ほどタイトルである可能性が高いので図７５のタイトル候補出力処理においては、まず文字列矩形を表の左上に近い順に並び替える。そして、その順番やステップＳ１４９で求めた文字数等の情報を使用して、文字列矩形の表内タイトルらしさの優先順位を決め、その順にタイトル候補として出力する。

【０１８９】優先順位の付け方としては、大きく分けて次の３通りが考えられる。

（ｗ）表の左上に近い順に優先順位を付ける。

（ｘ）隣りの文字列矩形内の文字矩形の文字数を調べ、その関係をもとに優先順位を決める。表内のタイトルには、「題名」や「表題」のようにタイトルであることを示す項目名がタイトルの左（または上）の位置にある場

合が多い。このような項目名とタイトルの関係は、それらの文字数を用いて表すことができる。例えば、２文字から数文字程度の文字列の右側（または下側）に、数文字から十数文字程度の文字列がある場合に、項目名とタイトルのペアがあると判断することができる。そこで、そのようなペアについて、上から順に優先順位を付ける。

（ｙ）一定の文字数の条件、または隣りの文字列矩形との間の一定の文字数の関係を満足するものだけを対象にして、表の左上に近い順に優先順位を付ける。

【０１９０】この場合は、表内の文字列矩形を左上に近い順に調べていき、文字列矩形内の文字数の合計が閾値以上であれば、その文字列矩形を項目候補とする。さらに、その文字列矩形の右側に他の文字列矩形があれば、その文字列矩形内の文字数にかかわらず、それをタイトル候補とする。

【０１９１】これは、元々１つの欄に項目とタイトルが併記されており、「項目：タイトル」のように１つの文字列矩形に両方の要素が含まれる場合を救済するためである。また、文字数の大きな文字列矩形は、それだけで表内タイトルらしいといえる。このような文字列矩形は項目候補として出力された場合でも、文字認識の結果からタイトルらしいと考えられれば、タイトルとして使用することができる。

【０１９２】文字列矩形内の文字数が閾値未満の場合は、その右側に他の文字列矩形があり、かつ、その中の文字数が閾値以上の場合に、前者を項目候補、後者をタイトル候補とする。

【０１９３】上記（ｗ）、（ｘ）、（ｙ）の各方法について、２０種類の文書画像を用いて実験した結果、（ｙ）の方法の場合に表内タイトルが候補の上位に入りやすく、これが最も抽出性能が良いことが分かった。そこで、図７５の処理では（ｙ）の方法に従って優先順位を決めている。

【０１９４】図７５において処理が開始されると、プロセッサ１４は、まず図７６に示すような各表内文字列矩形の左上頂点の座標（ｘ１，ｙ１）を用いて、ｘ１＋ｙ１の値の小さい順に、それらの文字列矩形をソートする（ステップＳ２２１）。そして、表内文字列矩形の集合をＳとし（ステップＳ２２２）、Ｓ内でｘ１＋ｙ１の値が最も小さいものをカレント矩形ＳＳとする（ステップＳ２２３）。

【０１９５】次に、ＳＳ内の文字矩形の文字数の合計が閾値TITLEMOJISUU以上かどうかを判定する（ステップＳ２２４）。例えば、TITLEMOJISUU＝７とする。ＳＳの文字数がTITLEMOJISUU以上であれば、ＳＳの右側に他の文字列矩形があるかどうかを調べる（ステップＳ２２５）。右側に文字列矩形がなければ、カレント矩形ＳＳを項目候補として出力し（ステップＳ２２６）、ＳＳがＳ内の最後の文字列矩形かどうかを判定する（ステップＳ２２７）。最後の文字列矩形でなければ、ＳＳの次にｘ１＋ｙ１の値の小さな文字列矩形を新たにＳＳとし（ステップＳ２３１）、ステップＳ２２４以降の処理を繰り返す。

【０１９６】ステップＳ２２５においてＳＳの右側に文字列矩形がある場合は、カレント矩形ＳＳを項目候補、その右側の文字列矩形をタイトル候補として出力し（ステップＳ２３０）、ステップＳ２２７以降の処理を行う。

【０１９７】また、ステップＳ２２４においてＳＳ内の文字数がTITLEMOJISUU未満の場合は、ＳＳの右側に他の文字列矩形があるかどうかを調べる（ステップＳ２２８）。右側に文字列矩形があれば、その文字数の合計がTITLEMOJISUU以上かどうかを判定する（ステップＳ２２９）。そして、それがTITLEMOJISUU以上であれば、ステップＳ２３０以降の処理を行う。

【０１９８】ステップＳ２２８においてＳＳの右側に文字列矩形がない場合、および、ステップＳ２２９において右側の文字列矩形の文字数がTITLEMOJISUU未満の場合は、ステップＳ２２７以降の処理を行う。そして、ステップＳ２２７においてＳＳが最後の文字列矩形であれば、処理を終了する。

【０１９９】このタイトル候補出力処理によれば、次の３つの場合に該当する文字列矩形が項目またはタイトル候補として出力される。

（α）カレント矩形の文字数が閾値以上で、その右側に文字列矩形がない場合、カレント矩形を項目候補として出力する。

（β）カレント矩形の文字数が閾値以上で、その右側に文字列矩形がある場合、カレント矩形を項目候補、右側の文字列矩形をタイトル候補として出力する。

（γ）カレント矩形の文字数が閾値未満で、その右側の文字列矩形の文字数が閾値以上である場合、カレント矩形を項目候補、右側の文字列矩形をタイトル候補として出力する。

【０２００】図７７は、こうして抽出された表内タイトルの第１候補を示している。図７７において、文字列矩形１１１は項目候補であり、文字列矩形１１２はタイトル候補である。このような表内タイトル抽出処理によれば、様々な表を含んだ文書画像に対しても、特別な操作や辞書等を用いずに、表内の項目およびタイトルの領域を抽出することができる。

【０２０１】こうして抽出された項目候補およびタイトル候補の文字列矩形は、図３のステップＳ１１と同様の処理により文字列として認識される。このとき、実際には、項目候補として抽出された文字列がタイトル文字列を含んでいる場合もあるので、認識結果のうち適当な部分を項目名あるいはタイトルとして使用する。

【０２０２】本発明においては、文字領域および文字列領域の形状は必ずしも矩形でなくてもよく、直線または

曲線により囲まれた任意の形状の領域を用いることができる。

【０２０３】

【発明の効果】本発明によれば、様々な文書画像に対して、特別な操作を行ったり、辞書等を用意したりしなくても、タイトル、宛先、発信元情報に相当する領域を容易に抽出することができる。これにより、画像データから抽出した文字列等をその画像のキーワードとして用いることもできるようになる。

【図面の簡単な説明】

【図１】本発明の原理図である。

【図２】システム構成図である。

【図３】文書画像のタイトル抽出処理のフローチャートである。

【図４】文書画像データを示す図である。

【図５】文字列抽出処理のフローチャートである。

【図６】ラベリング処理後の外接矩形を示す図である。

【図７】高さのヒストグラムを示す図である。

【図８】高さの最頻値を求めるためのヒストグラムを示す図である。

【図９】矩形高さテーブルを示す図である。

【図１０】矩形高さテーブルの内容に対応するヒストグラムを示す図である。

【図１１】大きな矩形から抽出された線分矩形を示す図である。

【図１２】部分線分矩形を示す図である。

【図１３】連結した部分線分矩形を示す図である。

【図１４】枠矩形を示す図である。

【図１５】重複している外接矩形を示す図である。

【図１６】ネストしている外接矩形を示す図である。

【図１７】二等辺三角形のヒストグラムを示す図である。

【図１８】重複・ネスト除去後の外接矩形を示す図である。

【図１９】矩形間の連結関係を示す図である。

【図２０】連結関係表を示す図である。

【図２１】文字列矩形を示す図である。

【図２２】文字列矩形の抽出処理を示す図である。

【図２３】抽出された文字列矩形を示す図である。

【図２４】文字列矩形加工処理のフローチャートである。

【図２５】ノイズ除去後の文字列矩形を示す図である。

【図２６】文字列矩形の統合処理を示す図である。

【図２７】統合された文字列矩形を示す図である。

【図２８】文書領域を示す図である。

【図２９】下線矩形を示す図である。

【図３０】枠付き・罫線・下線チェック後の文字列矩形を示す図である。

【図３１】線分抽出処理のフローチャートである。

【図３２】ワイルドカードがある場合の線分矩形を示す図である。

【図３３】ワイルドカードを示す図である。

【図３４】線分抽出処理のコードを示す図（その１）である。

【図３５】線分抽出処理のコードを示す図（その２）である。

【図３６】線分抽出処理のコードを示す図（その３）である。

【図３７】線分抽出処理の詳細フローチャート（その１）である。

【図３８】線分抽出処理の詳細フローチャート（その２）である。

【図３９】線分抽出処理の詳細フローチャート（その３）である。

【図４０】タイトル・宛先・発信元抽出処理のフローチャートである。

【図４１】オーバラップしている文字列矩形を示す図である。

【図４２】第１の宛先抽出処理のフローチャートである。

【図４３】第２の宛先抽出処理のフローチャートである。

【図４４】タイトルと宛先／発信元の第１の配置を示す図である。

【図４５】タイトルと宛先／発信元の第２の配置を示す図である。

【図４６】タイトルと宛先／発信元の第３の配置を示す図である。

【図４７】タイトルと宛先／発信元の第４の配置を示す図である。

【図４８】複数の宛先／発信元を示す図である。

【図４９】タイトルおよび宛先・発信元情報の抽出結果を示す図である。

【図５０】タイトルおよび宛先・発信元情報の他の抽出結果を示す図である。

【図５１】表形式文書を示す図である。

【図５２】表内タイトル抽出処理のフローチャートである。

【図５３】表形式文書の画像データを示す図である。

【図５４】表形式文書のラベリング結果を示す図である。

【図５５】表矩形抽出処理のフローチャートである。

【図５６】表形式文書の文字列矩形を示す図である。

【図５７】第１の文字列分割処理のフローチャートである。

【図５８】文字列矩形内の文字矩形の順位を示す図である。

【図５９】ソート後の文字矩形の順位を示す図である。

【図６０】縦罫線を含む文字列矩形を示す図である。

【図６１】分割された文字列矩形を示す図である。

43

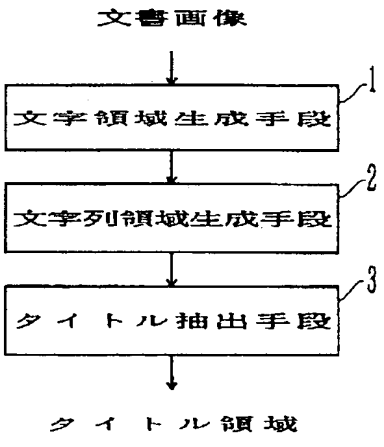
【図６２】第２の文字列分割処理のフローチャート（その１）である。
【図６３】第２の文字列分割処理のフローチャート（その２）である。
【図６４】文字列矩形内のラベリング結果を示す図である。
【図６５】分割処理後の文字列矩形を示す図である。
【図６６】文字矩形と文字数の関係を示す図である。
【図６７】表矩形内の表外文字列矩形を示す図である。
【図６８】表矩形内の文字列矩形を示す図である。
【図６９】上罫線チェック処理のフローチャートである。
【図７０】表外文字列矩形除去処理のフローチャートである。
【図７１】第１の探索範囲を示す図である。
【図７２】第２の探索範囲を示す図である。
【図７３】第３の探索範囲を示す図である。
【図７４】表外文字列矩形除去後の文字列矩形を示す図である。
【図７５】タイトル候補出力処理のフローチャートである。
【図７６】文字列矩形の左上頂点の座標を示す図である。
【図７７】表内タイトルの抽出結果を示す図である。

44

【符号の説明】
１　文字領域生成手段
２　文字列領域生成手段
３　タイトル抽出手段
１１　電子ファイリング装置
１２　光電変換装置
１３　ディスプレイ端末
１４　プロセッサ
１５　メモリ
１６　バス
２１、２２、２３、２４、２５、２６、２７、３１、３２、３３、３４、３５、６１、６２、６３、６４、６５　外接矩形
３６、３７、３８、３９、４０　二等辺三角形
４１、４２、４３　二等辺三角形のヒストグラム
５１、５２、５３、５４、５５、５６、５７、５８　ポインタ
７１、８１、８２、８３、９１、１０１、１０８、１０９、１１１、１１２、１１３、１１４、１１５、１１６、１１７、１２２、１２３、１３１　文字列矩形
７２　下線矩形
８０、１２１　表矩形
９２、９３、９４、９５、１０２、１０３、１０４、１０５、１０６、１０７、１１０　文字矩形

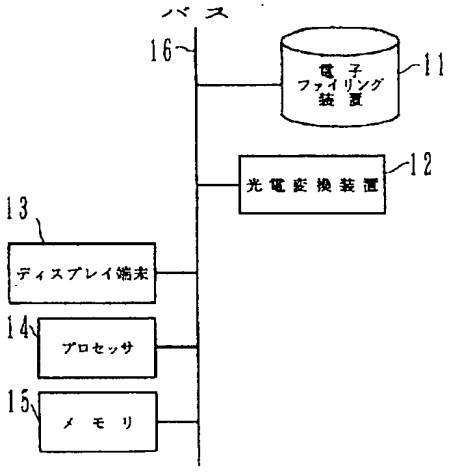
【図１】

本発明の原理図



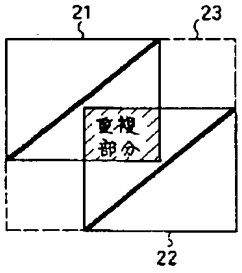
【図２】

システム構成図



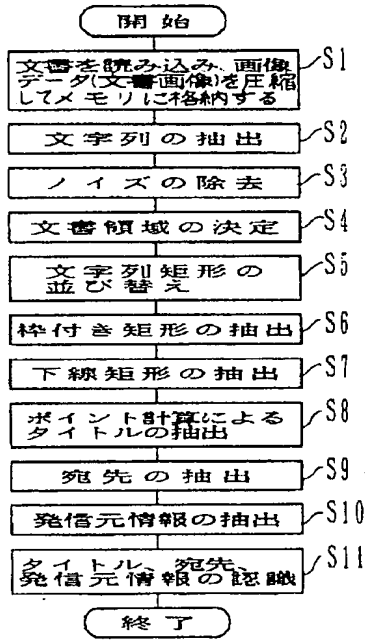
【図１５】

重複している外接矩形を示す図



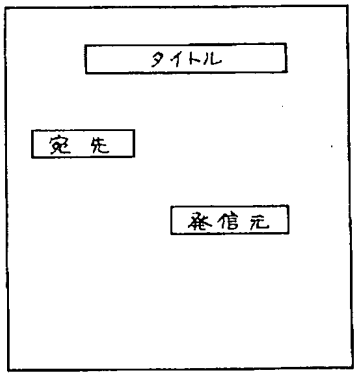
【図３】

文書画像のタイトル抽出処理のフローチャート



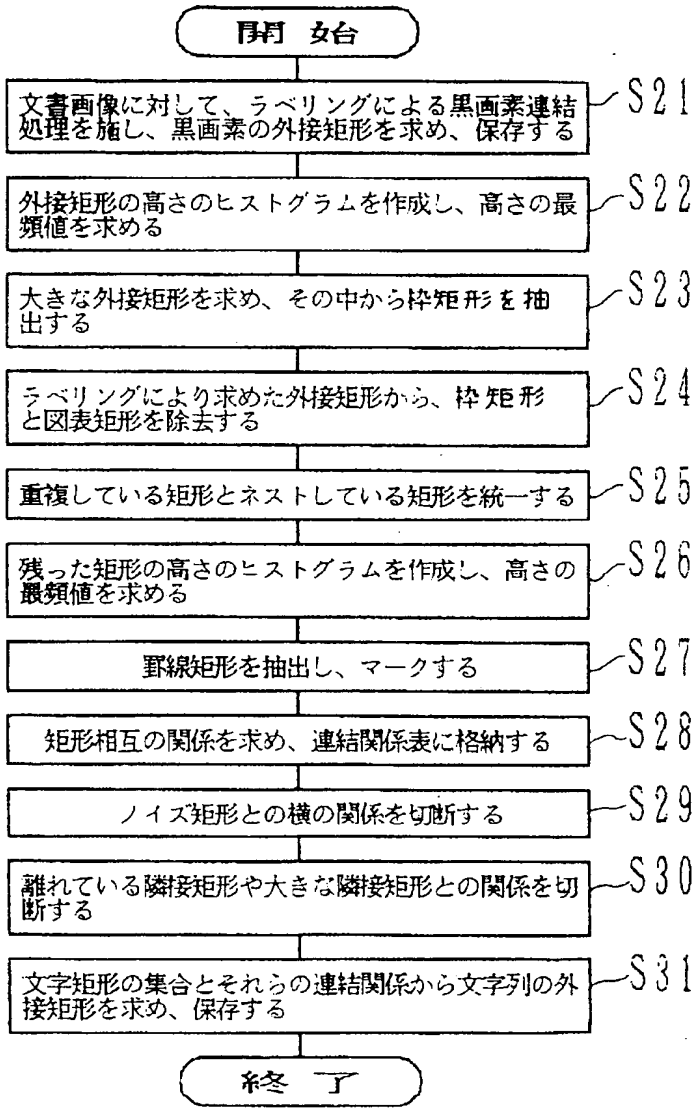
【図４４】

タイトルと宛先/発信元の第1の配置を示す図



【図５】

文字列抽出処理のフローチャート



【図４】

文書画像データを示す図

ソフトウェア販推レポート　送付表　[１８]

新横浜

[ＴＥＣＨビル]
オープンシステム事業本部
（オープンソフトウェア事業部）
ソフトウェア管理部長殿　第一開発部長殿　第二開発部長殿　第三開発部長殿　技術部長殿
（オープン標準推進統括部）アーキテクティ部長殿　事務支援部長殿
[日高不動産ビル：５５Ｃ]
ミドルウェア事業本部
（ビジネス推進統括部）
開発支援部長殿　ＩＳＶサポートセンター長殿　よろず相談室長殿
情報処理事業推進本部
（ソフトウェア販路支援統括部）ソフトウェア販売管理部長殿
情報システム事業本部
（ソフトウェア事業部）第六開発部長殿

川崎工場

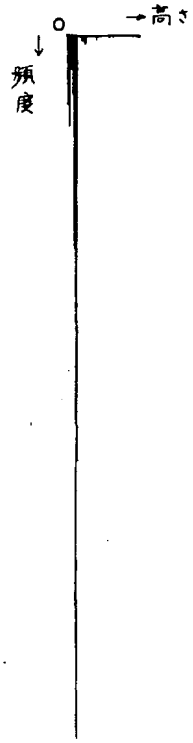
社内システム開発本部
（電算情報システム統括部）システム運用管理部長殿
（生産管理システム統括部）生産情報システム支援担当部長殿　資材管理システム部長殿
販売情報連携システム本部
（システム管理部）計画部長殿
（ネットワークシステム推進部）システム第一部長殿
（ＥＣ＆Ｅシステム統括部）第二システム部長殿
（金融システム統括部）第一システム部長殿　第二システム部長殿
（第一ネットワークシステム統括部）
（プロジェクト推進部）第一ネットワーク部長殿　第二ネットワーク部長殿　第三ネットワーク部長殿
（第二ネットワークシステム統括部）
第一ネットワーク部長殿　第二ネットワーク部長殿　第三ネットワーク部長殿
情報処理事業推進本部
（事業管理部）管理課長殿
（営業管理部）第二工場部長殿
（ソフトウェア開発支援推進課）ソフトウェア計画部長殿
（技術支援部）特定地区担当部長殿
ミドルウェア事業本部
（企画部）開発推進課長殿　計画部長殿
（第二ミドルウェア事業本部）第一開発部長殿　第三開発部長殿　グローバル開発部長殿
電子ファイリングプロジェクト部長殿
（第三ミドルウェア事業本部）第一開発部長殿　第二開発部長殿
情報システム事業本部
（本社本部）電算機開発部長殿　スーパーコンピュータ開発課長殿
電算第一技術部長殿　電算機第二技術部長殿　電算機第三技術部長殿　電算機第四技術課長殿
オープンシステム事業本部
（オープンシステム事業部）
第一開発部長殿　第二技術部長殿　第三技術部長殿　第五技術部長殿　第六技術課長殿　技術技術部長殿
パーソナルビジネス本部
（ＰＣ－ＶＡＮ－ＥＸ推進課）ネストマサポート部長殿
（ファイル開発統括部）マニュアル制作課長殿
[富士総研究所）情報システムセンター技術管理殿
（テクニカルセンター）富士通カストマエンジニアリング（ネットワーク開発部）企画調整部長殿
輸出管理本部
第一輸出管理室長殿
海外事業本部
（海外関連販売事業部×１　東南アジア）国際販売課課長殿

受付

[１９]

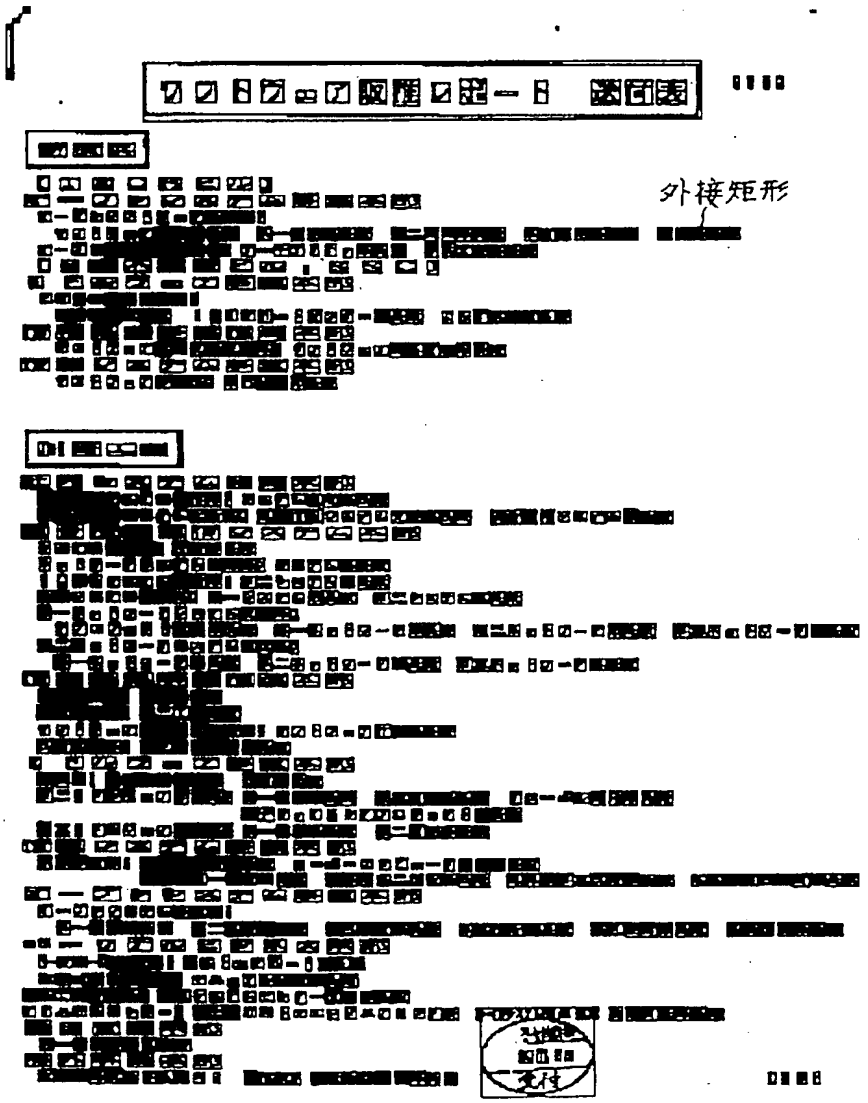
【図７】

高さのヒストラムを示す図



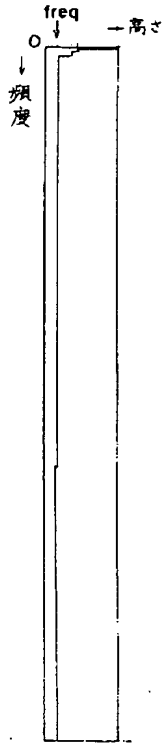
【図６】

ラベリング処理後の外接矩形を示す図



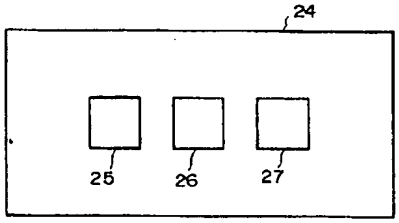
【図８】

高さの最頻値を求めるためのヒストグラムを示す図



【図１６】

ネストしている外接矩形を示す図



【図６６】

文字矩形と文字数の関係を示す図

文字矩形　高さH　　文字数＝［Ｗ／Ｈ］
幅Ｗ

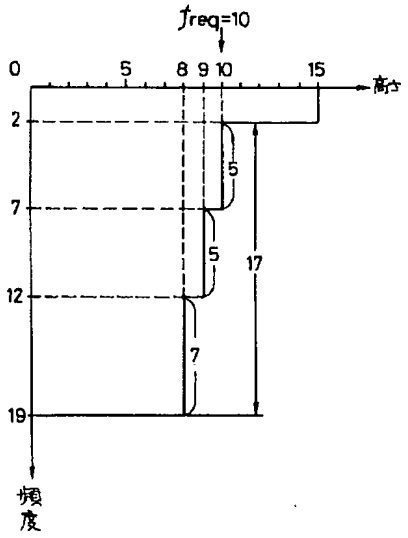
【図９】

矩形高さテーブルを示す図

頻　度	最大高さ
2	15
7	10
12	9
19	8

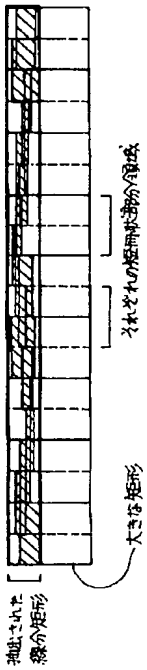
【図１０】

矩形高さテーブルの内容に対応するヒストグラムを示す図



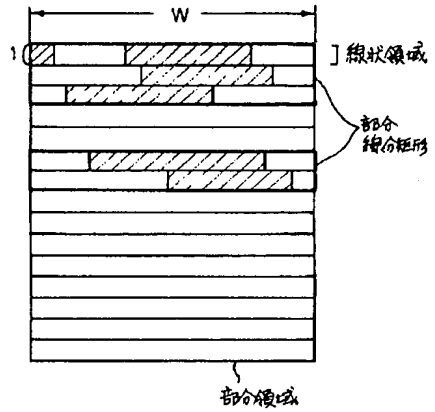
【図１１】

大きな矩形から抽出された線分矩形を示す図



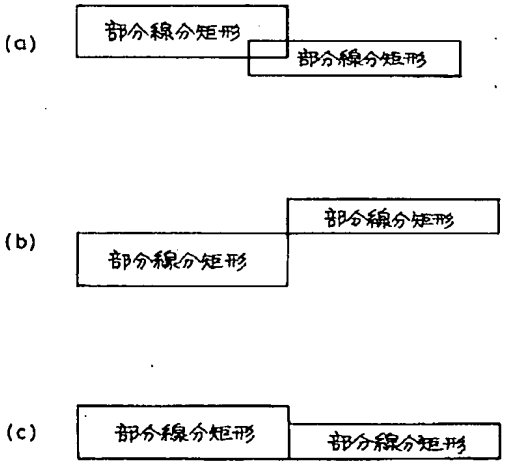
【図１２】

部分線分矩形を示す図



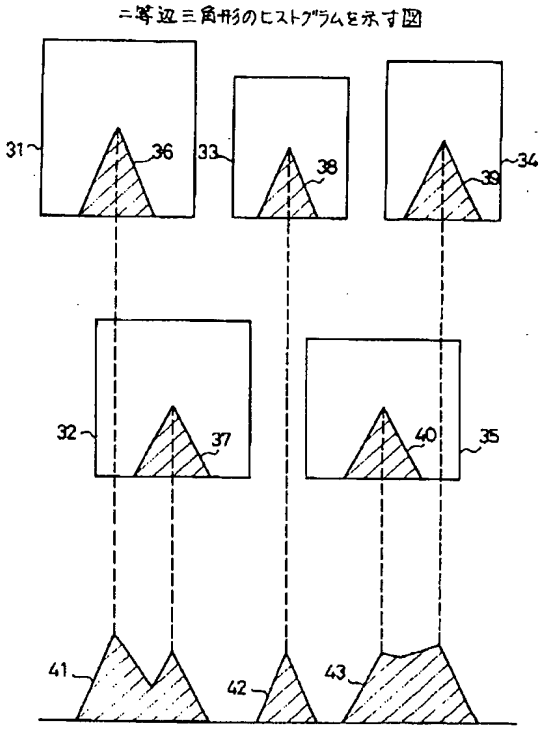
【図１３】

連結した部分線分矩形を示す図



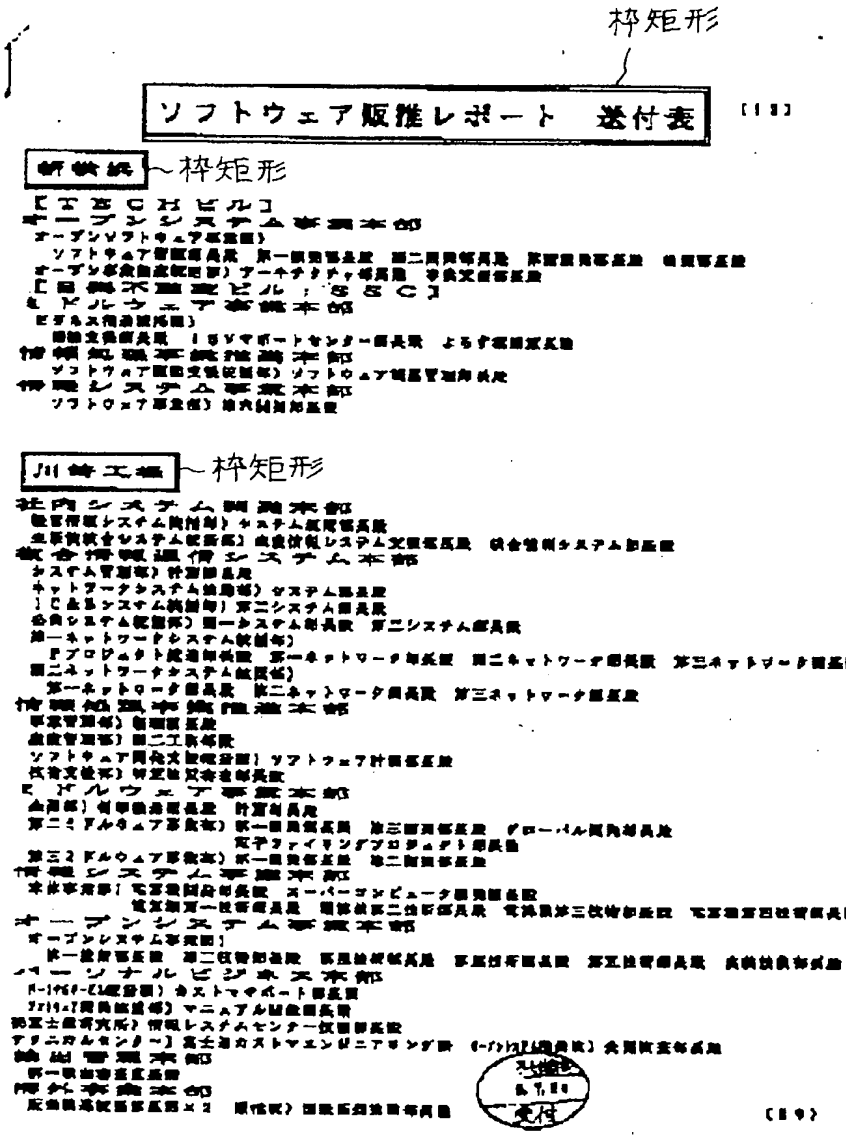
【図１７】

二等辺三角形のヒストグラムを示す図



【図１４】

枠矩形を示す図

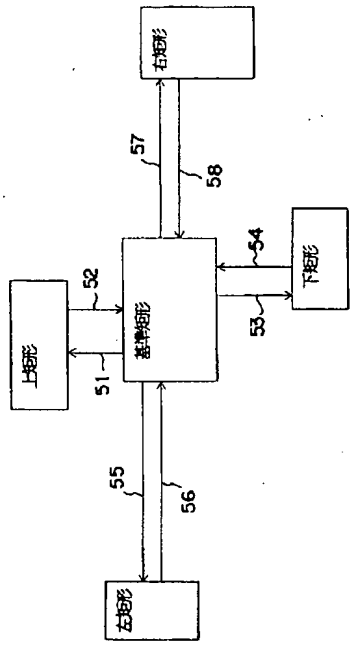


【図１８】

重複・ネスト除去後の外接矩形を示す図

【図１９】

矩形間の連結関係を示す図



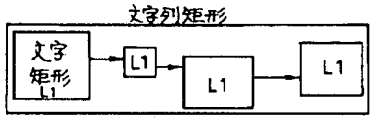
【図２０】

連結関係表を示す図

基準矩形のラベル値
上矩形へのポインタ
上矩形からのポインタ
下矩形へのポインタ
下矩形からのポインタ
左矩形へのポインタ
左矩形からのポインタ
右矩形へのポインタ
右矩形からのポインタ

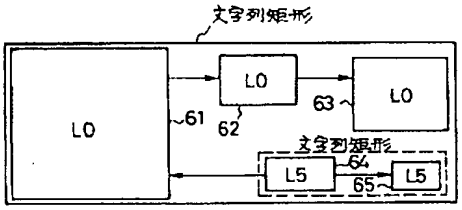
【図２１】

文字列矩形を示す図



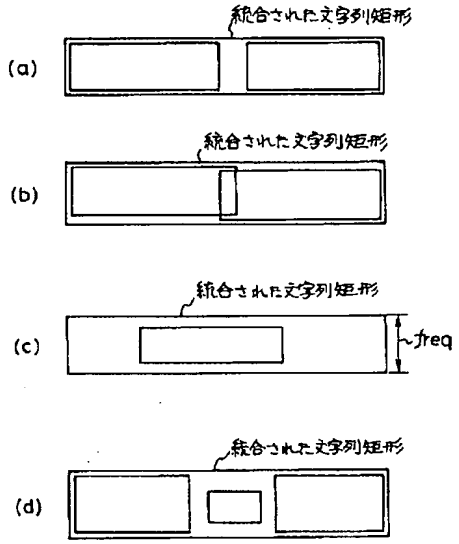
【図２２】

文字列矩形の抽出処理を示す図



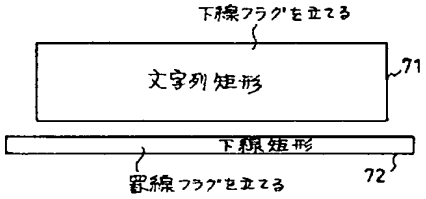
【図２６】

文字列矩形の統合処理を示す図



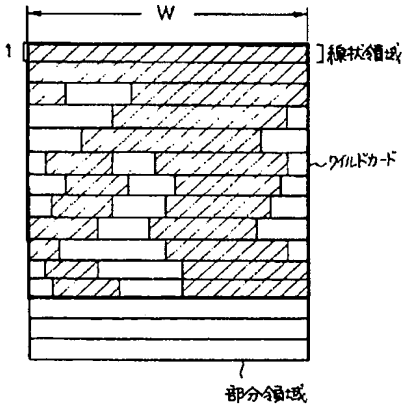
【図２９】

下線矩形を示す図



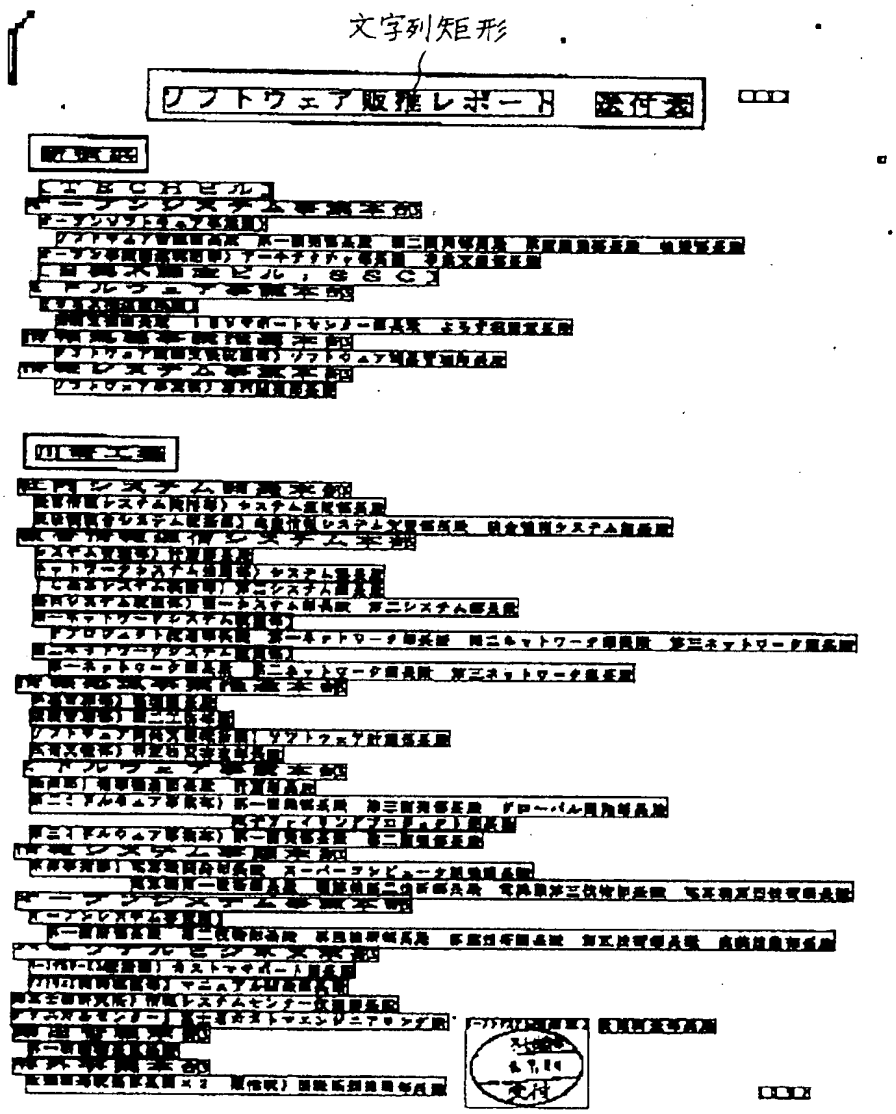
【図３３】

ワイルドカードを示す図



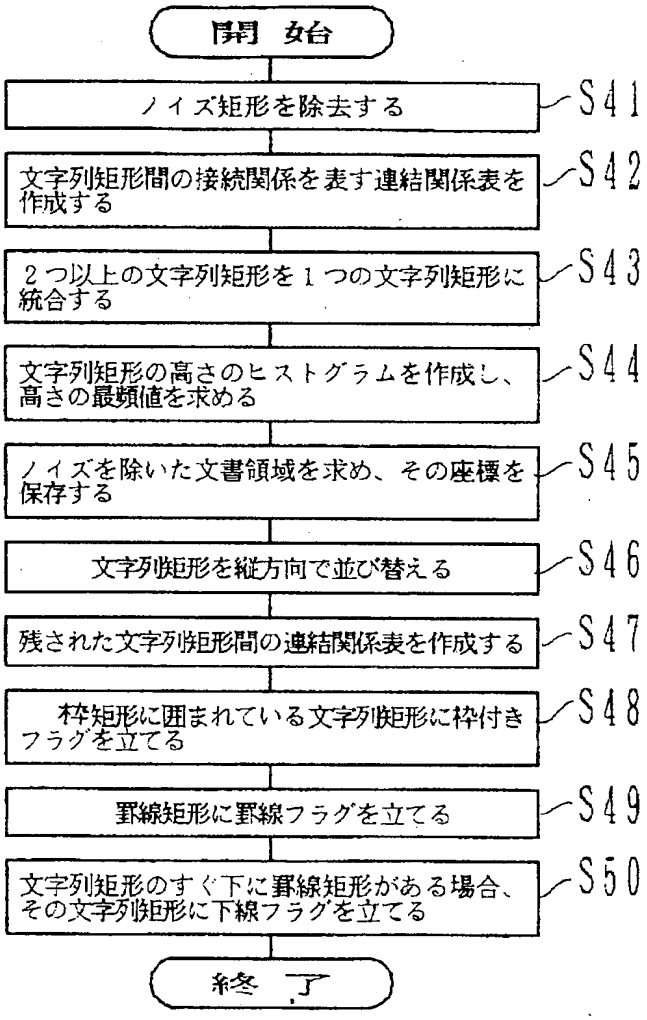
【図２３】

抽出された文字列矩形を示す図



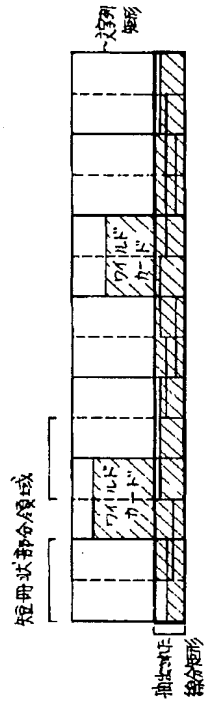
【図２４】

文字列矩形加工処理のフローチャート



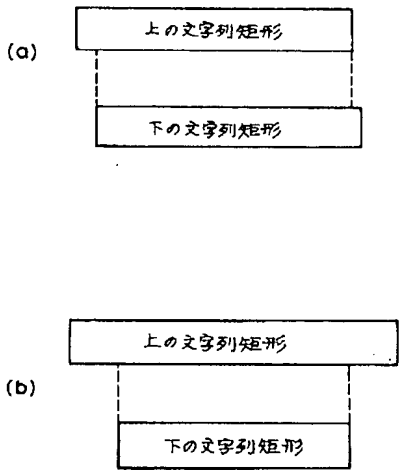
【図３２】

ワイルドカードがある場合の線分矩形を示す図



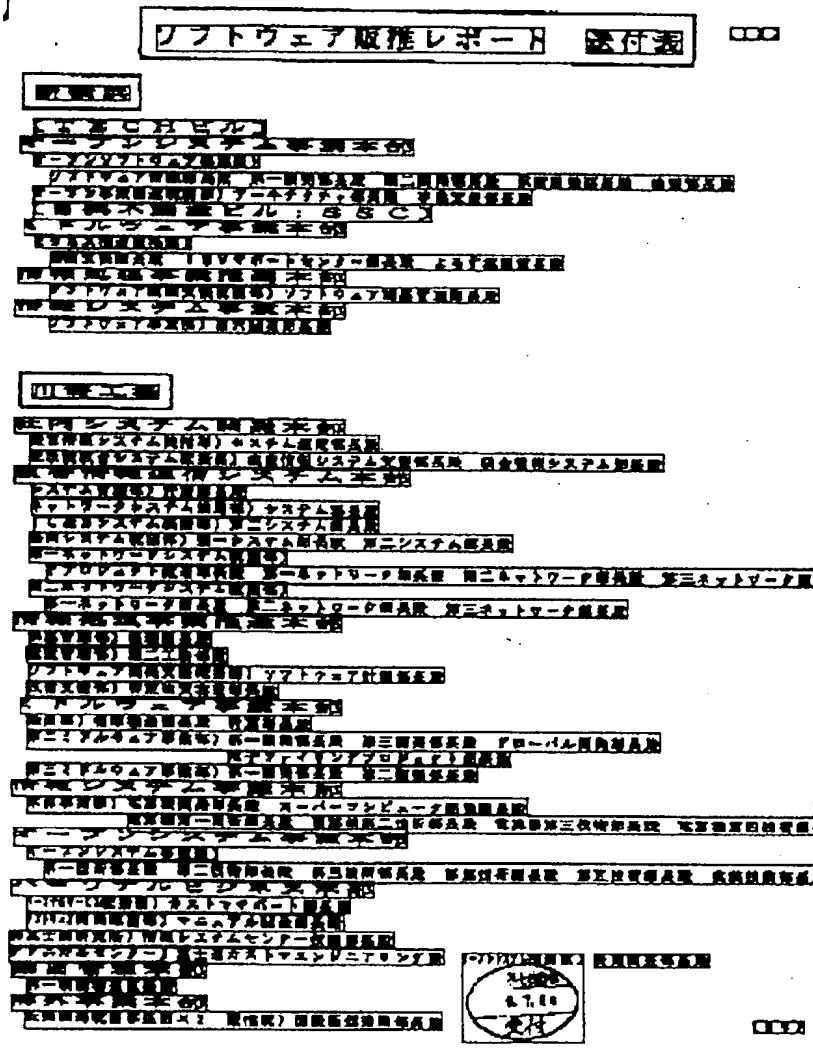
【図４１】

オーバラップしている文字列矩形を示す図



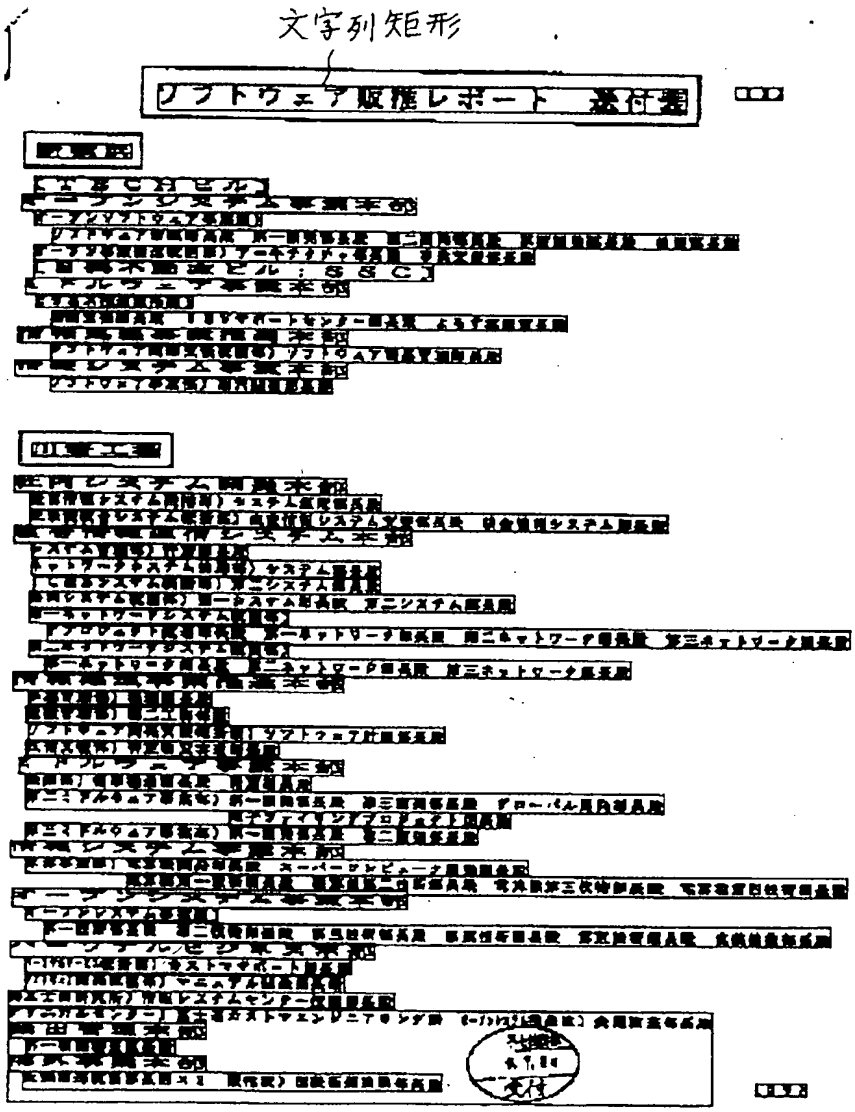
【図２５】

ノイズ除去後の文字列矩形を示す図



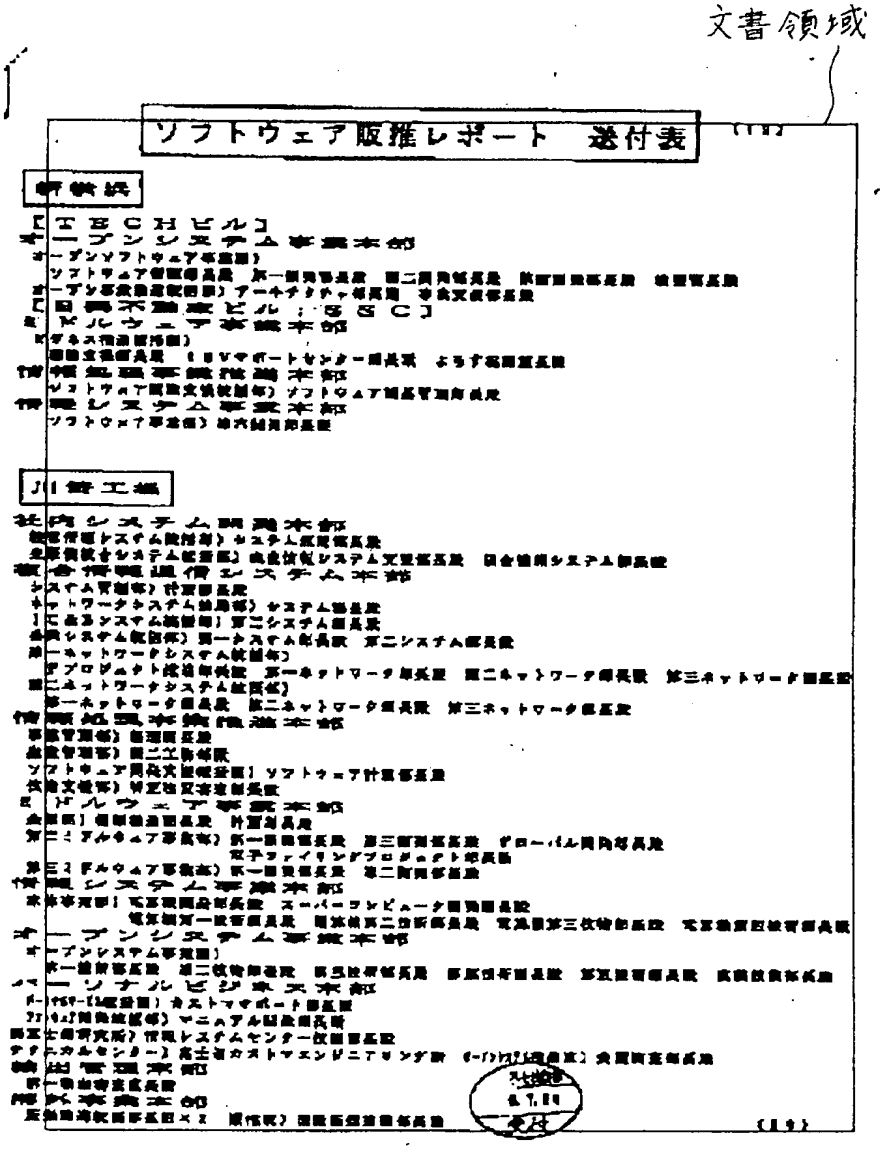
【図２７】

統合された文字列矩形を示す図

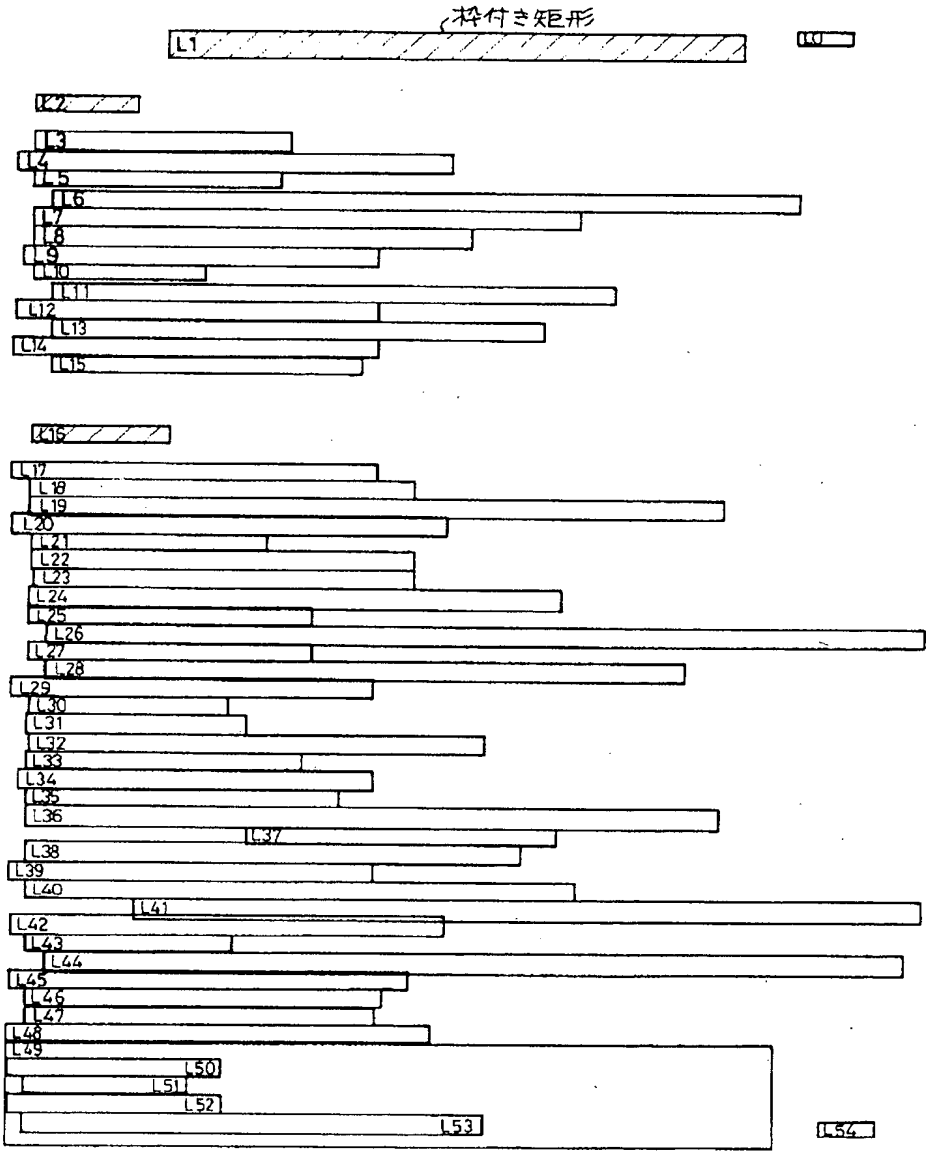


【図２８】

文書領域を示す図

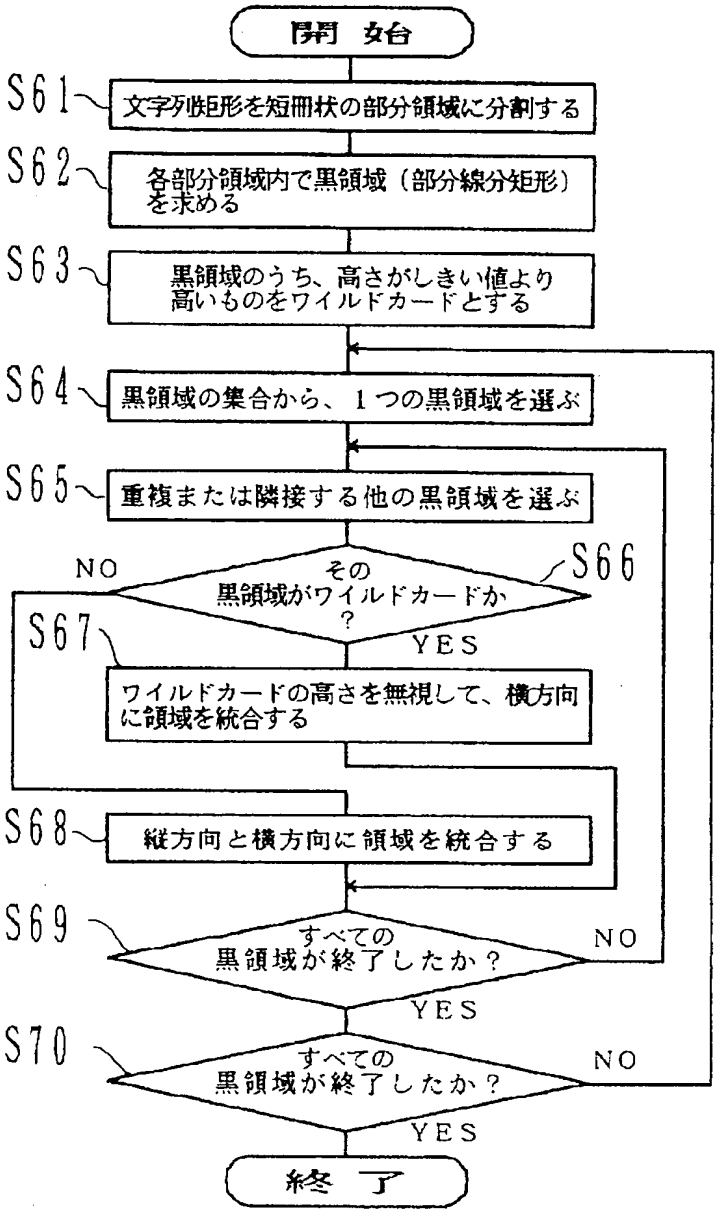


【図３０】


枠付き・罫線・下線チェック後の文字列矩形を示す図
枠付き矩形
L1
L0
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
L16
L17
L18
L19
L20
L21
L22
L23
L24
L25
L26
L27
L28
L29
L30
L31
L32
L33
L34
L35
L36
L37
L38
L39
L40
L41
L42
L43
L44
L45
L46
L47
L48
L49
L50
L51
L52
L53
L54

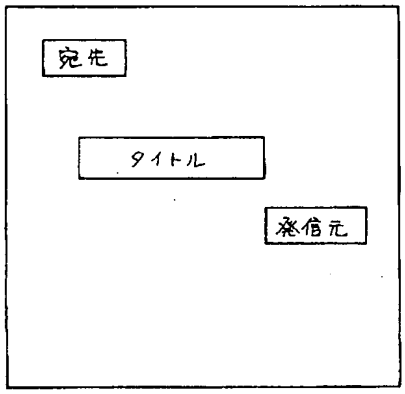
【図３１】

線分抽出処理のフローチャート



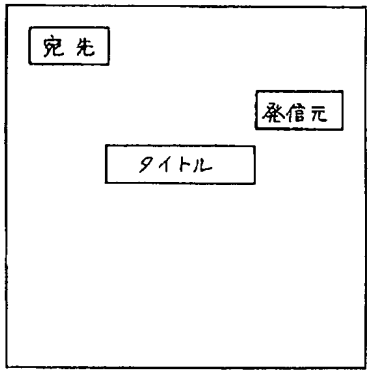
【図４５】

タイトルと宛先/発信元の第2の配置を示す図



【図４６】

タイトルと宛先/発信元の第3の配置を示す図



【図３４】

線分抽出処理のコードを示す図（その１）

```
先ず、部分矩形の中で文字列矩形の高さ×0.3以上のものをワイルドカード
部分矩形としてマークする(use=9)。
それ以外は、普通の矩形としてマークする(use=0)。

lnum=0

部分矩形全部に渡って以下を処理。カレント矩形：i {

  if( (矩形iのuse が 0) または (矩形iのuseが9) ) {
    xlf =              矩形iの左端座標
    xr  =              矩形iの右端座標
    yup = line__starty = 矩形iの上端座標
    ybl = line__endy   = 矩形iの下端座標

    if(矩形iのuse が 0) {
      standard__st=yup;
      standard__en=ybl;
      standard__h=ybl-yup+1;
      b__use=0; /* 最初の線分がwildcardでなく、standardが設定あり */
      height=ybl-yup+1;
      矩形iのuse = 1;
    }
    else { /* use:9の場合 */
      standard__st=0;
      standard__en=0;
      standard__h=0;
      b__use=9; /* 最初の線分がwildcardで、standardが設定なし */
      height2=ybl-yup+1;
      height=0;
    }
    部分矩形全部に渡って以下を処理。カレント矩形：k {
C1→ α
C2→ β
    }
    /* 全ての線分がwildcard線分だった場合 */
    if( (b__use が 9) ) {
      /* 最初の線分の高さを長い線分の高さとする */
      height=height2;
    }

    求めた線分座標(左端：xlf, 右端：xr, 上端：line__starty, 下端：line__endy)を
    yokolineのlnum番目に格納

    lnumを１つインクリメント
  }
}
```

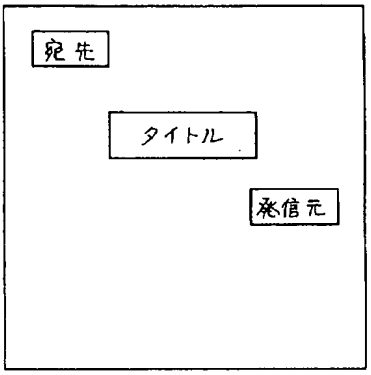
【図３５】

線分抽出処理のコードを示す図（その２）

```
rxlt = 矩形kの左端座標
rxr  = 矩形kの右端座標
ryup = 矩形kの上端座標
rybl = 矩形kの下端座標
rheight=rybl-ryup+1;

/* standardの値が設定されている */
if( (b__use が 0) ) {
  /* standardありで、右の矩形がwildcard */
  if(矩形kのuseが9) {

    /* 右隣の矩形が横&縦方向に1dot以上の重なり */
    if( ((xr+1)>=rxlf) && (xr < rxr) &&
      ((ybl+1)>=ryup) && ((yup-1)<=rybl) ) {
      xr      = rxr;
    }
  }
  /* standardありで、右の矩形がwildcardではない */
  else if(矩形kのuseが0) {
  /* 右隣の矩形が横&縦方向に1dot以上の重なり */
  if( ((xr+1)>=rxlf) && (xr < rxr) &&
    ((ybl+1)>=ryup) && ((yup-1)<=rybl) &&
    (standard__h-TH__HEIGHTDOT <=rheight) &&
    (rheight <=standard__h+TH__HEIGHTDOT)  ) {
    矩形kのuse      = 2;
    xr      = rxr;
    yup     = ryup;
    ybl     = rybl;
    hei=rybl-ryup+1;
    if(hei>height) {
      height=hei;
    }
    if(ryup<line__starty)
      line__starty=ryup;
    if(rybl>line__endy)
      line__endy=rybl;

    standard__h=hei;
  }
 }
}
```

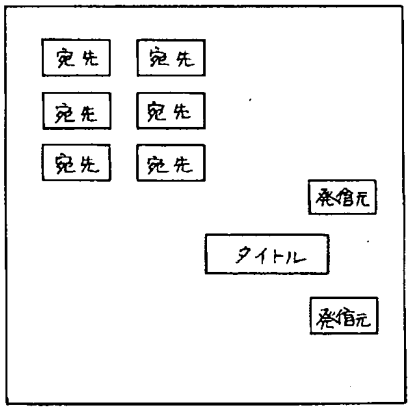
【図４７】

タイトルと宛先/発信元の第4の配置を示す図



【図４８】

複数の宛先/発信元を示す図



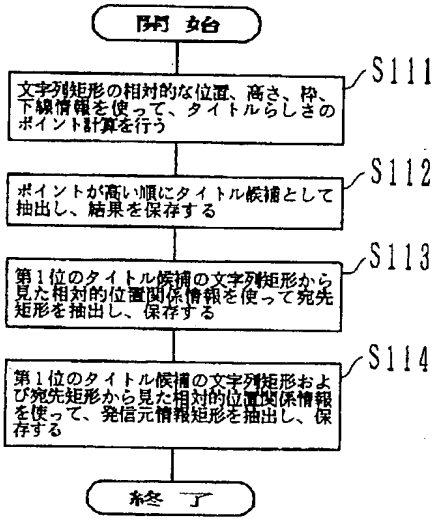
【図３６】

線分抽出処理のコードを示す図（その３）

```
/* standardの値が設定されていない */
else if((b_use == 9) ) {
  /* standardなしで、右の矩形がwildcard */
  if(矩形kのuseが9) {

    /* 右隣の矩形が横方向に1dot以上の重なり */
    if( ((xr+1)>=rx1f) && (xr < rxr) ) {
      xr      = rxr;
    }
  }
  /* standardなしで、右の矩形がwildcardではない */
  else if(矩形kのuseが0) {

    /* 右隣の矩形が横方向に1dot以上の重なり */
    if( ((xr+1)>=rx1f) && (xr < rxr) ) {

      b_use=0; /* standard設定済み */
      矩形kのuse      = 2;
      xr      = rxr;
      yup     = ryup;
      ybl     = rybl;
      hei=rybl-ryup+1;
      if(hei>height) {
        height=hei;
      }
      standard_st=ryup;
      standard_en=rybl;
      standard_h=hei;
      line_starty=ryup;
      line_endy=rybl;
    }
  }
}
```

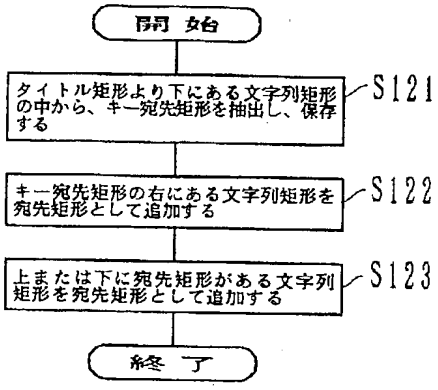
【図４０】

タイトル・宛先・発信元抽出処理のフローチャート



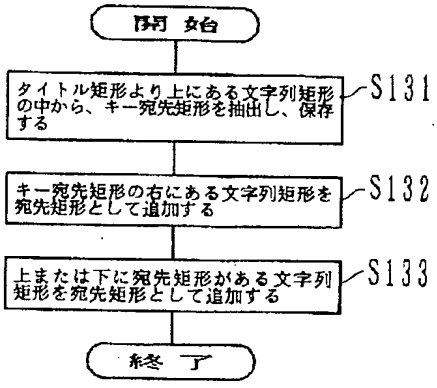
【図４２】

第１の宛先抽出処理のフローチャート



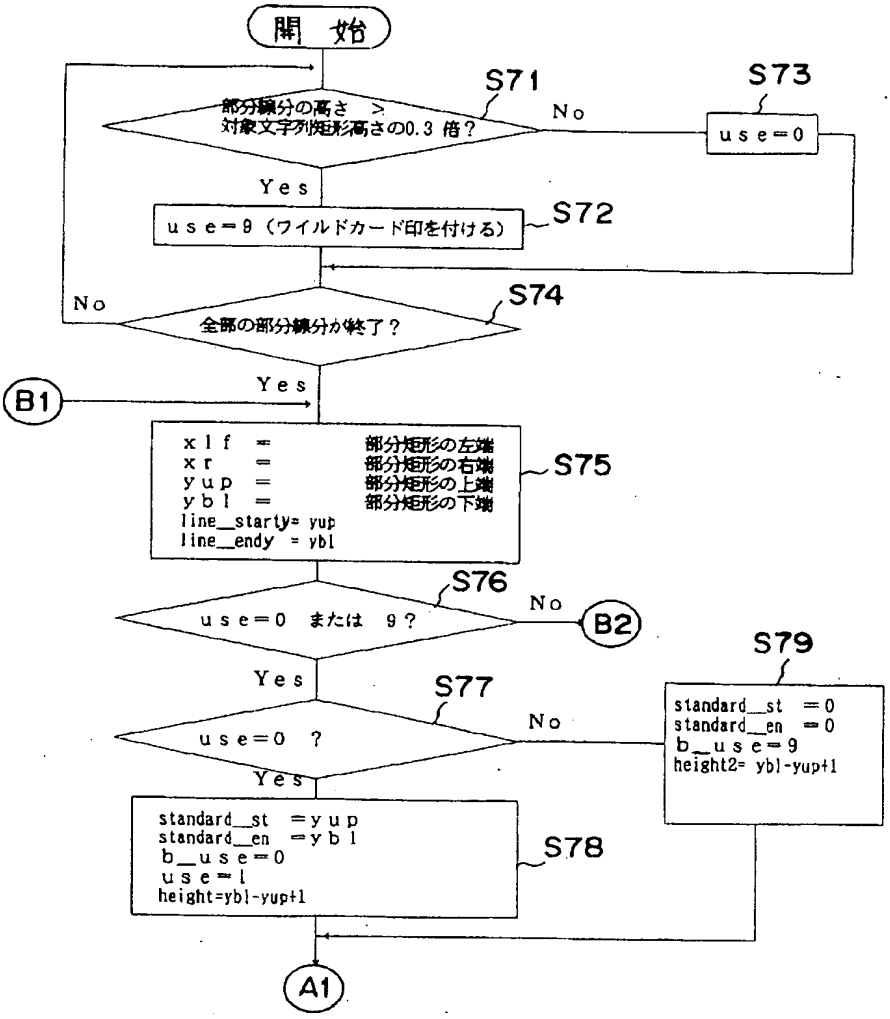
【図４３】

第２の宛先抽出処理のフローチャート



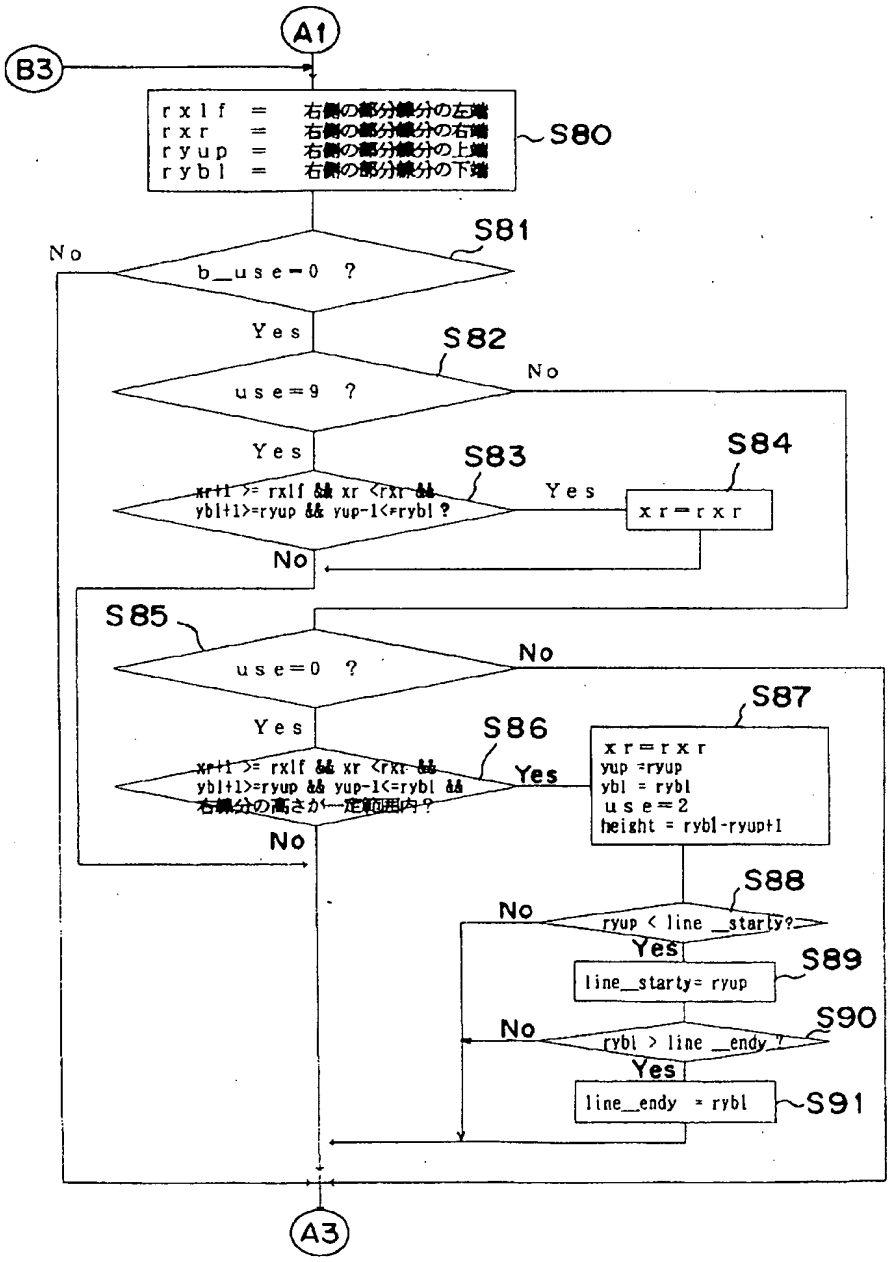
【図３７】

線分抽出処理の詳細フローチャート（その１）



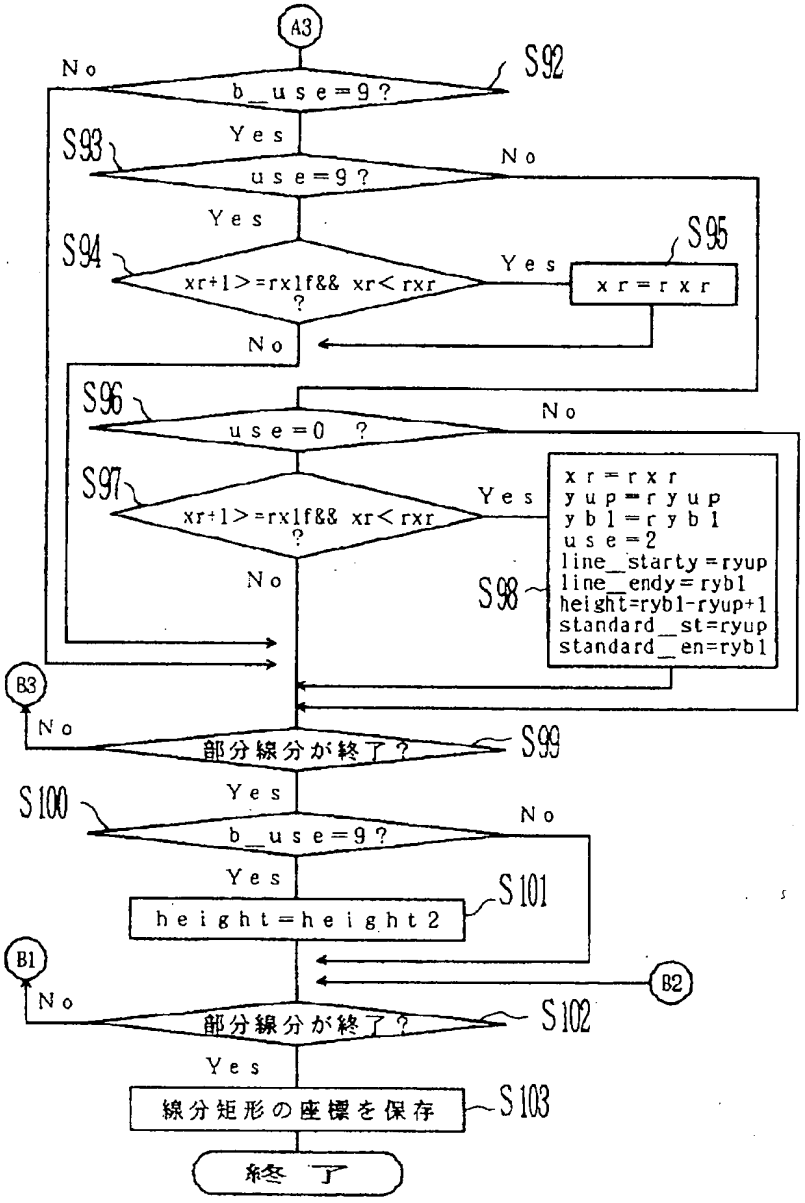
【図38】

線分抽出処理の詳細フローチャート(その2)



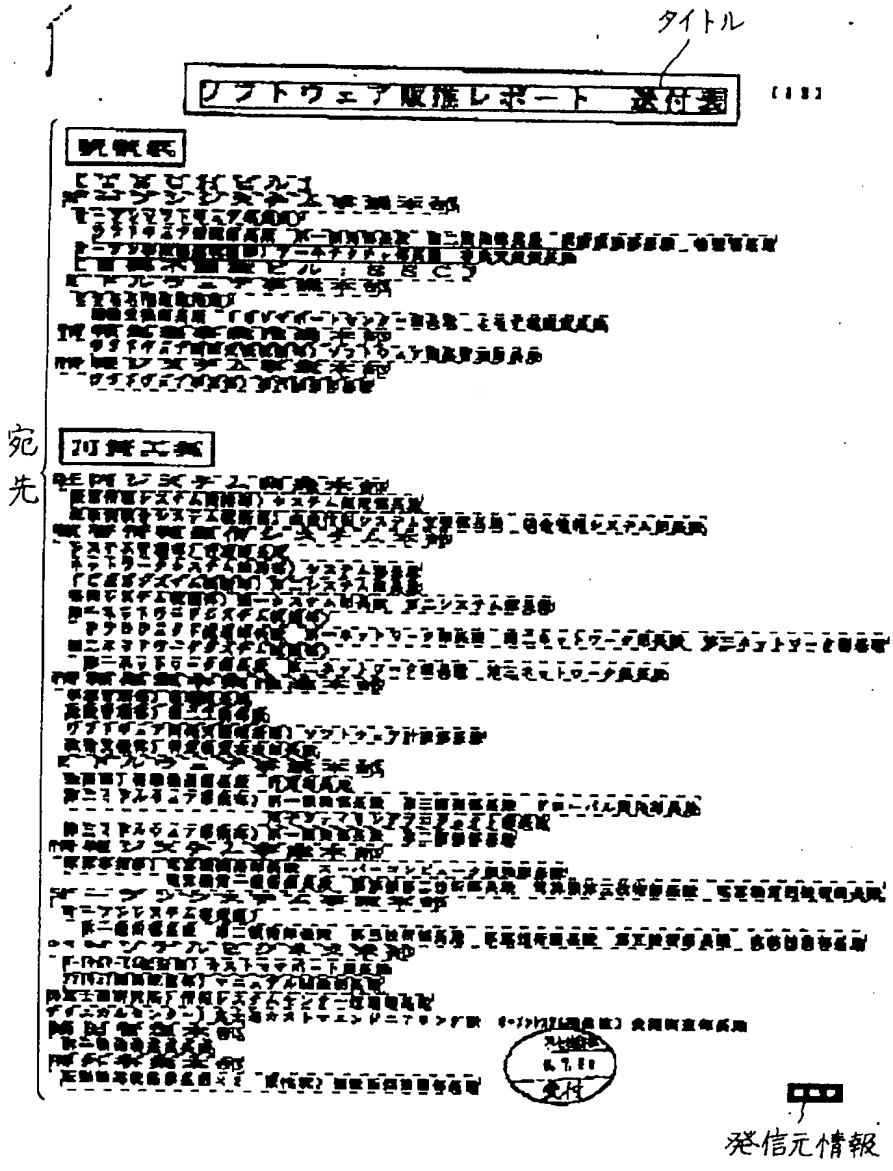
【図３９】

線分抽出処理の詳細フローチャート（その３）



【図４９】

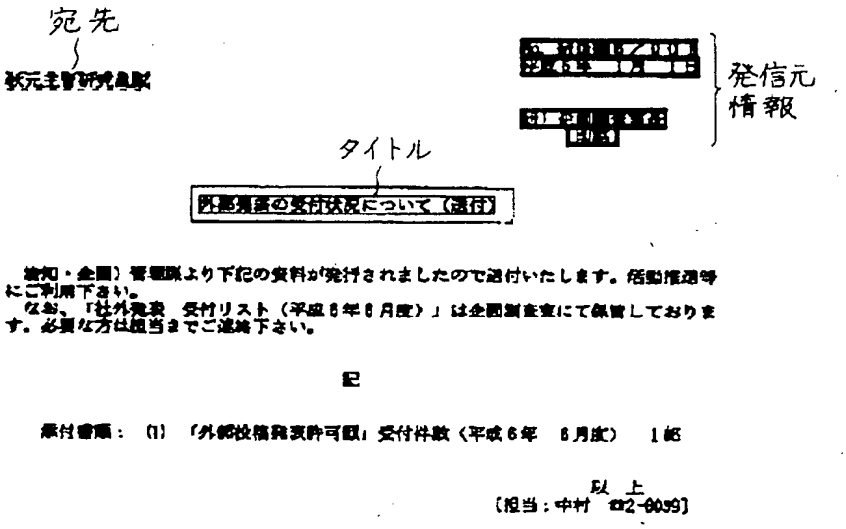
タイトルおよび宛先・発信元情報の抽出結果を示す図



【図５０】

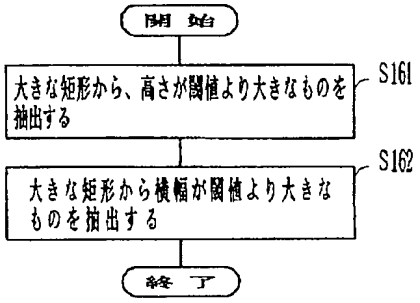
タイトルおよび宛先・発信元情報の他の抽出結果を示す図

technical_news



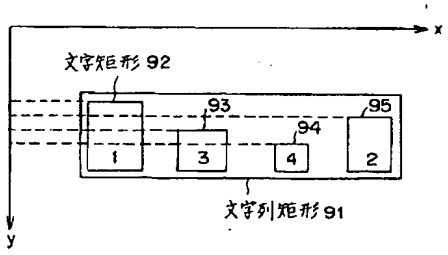
【図５５】

表矩形抽出処理のフローチャート



【図５８】

文字列矩形内の文字矩形の順位を示す図



【図51】

表形式文書を示す図

表罫線

出張報告書　　回覧　　1／18

表題	マルチメディア とパターン認識シンポジウム	発行No.：0123456789－1995 発行元：○○○事業部△△△部××××課 発行日：平成　7年12月31日	

報告者				公開範囲	1：事業部内 2：部門内 3：部内　4：課内
従業員番号	氏　名	従業員番号	氏　名		
1) 111111	勝山　裕	2)		報告書の性格	
3)		4)		1：普通　4：海外	①：速報
5)		6)		2：急ぎ　5：重要	
7)		8)		3：国内　6：その他	2：詳細
9)		10)			

要約（３５０文字）：パターン認識研究の中で、文字認識は、郵便番号読み取り装置のように最も明確なニーズをもった研究対象の一つである。画像をテキストコードに変換する文字認識技術はマルチメディア時代のさまざまな新サービスに利用されようとしている。ここではこれから文字認識を研究されようとする方に、現状のサービス形態、技術レベルについて概観する。文字認識方式にはオフライン型とオンライン型の２種類がある。紙に書かれた文字をスキャナで読み取って認識する方式はオフライン文字認識と呼ばれ、郵便番号読み取り装置、帳票ＯＣＲ、文書ＯＣＲ等がある。一方、最近話題を呼んだペン入力コンピュータによる文字認識方式はオンライン文字認識と呼ばれ、ペンの筆記過程を時々刻々コンピュータに取り込み、その情報をもとに手書き文字を認識する。

社内分類	独自分類	通番	関連出張報告書	関連番号	関連製品名
(1)1234	(1)234	(1)0001	(1)	(1)AAA01	(1)文書リーダ
(2)5678	(2)456	(2)	(2)	(2)BBB02	(2)帳票ＯＣＲ
(3)	(3)	(3)	(3)	(3)CCC03	(3)
		(4)	(4)	(4)DDD04	(4)
		(5)	(5)	(5)EEE05	(5)

キーワード			所属長印
	(1)ＯＣＲ	(2)文字認識	
	(3)画像処理	(4)ラベリング	
	(5)領域抽出	(6)電子ファイリングシステム	
	(7)文書構成要素	(8)タイトル	
	(9)表	00項目	
	01	02	
	03	04	受付印
	05	06	

関係配布先名	コード	関係配布先名	コード	
(1)鈴木課長殿	1111	(2)佐藤課長殿	2222	
(3)山本部長殿	3333	(4)加藤課長殿	4444	
(5)		(6)		
(7)		(8)		整理番号
(9)		00		
01		02		
03		04		
05		06		合計配布部数　４部
07		08		
09		00		

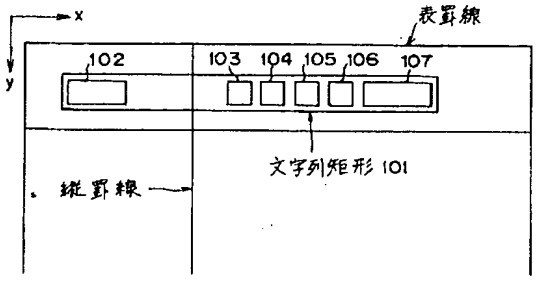
複写の発行元確認　①：要　2：不要　　連絡先・電話番号　氏名　勝山　裕　　☎777-1111

【図５２】

表内タイトル抽出処理のフローチャート

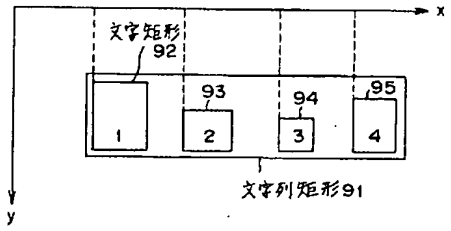
【図６０】

縦罫線を含む文字列矩形を示す図



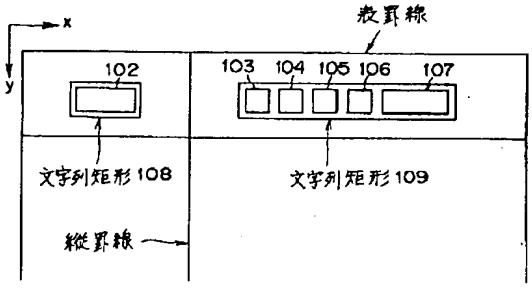
【図５９】

ソート後の文字矩形の順位を示す図



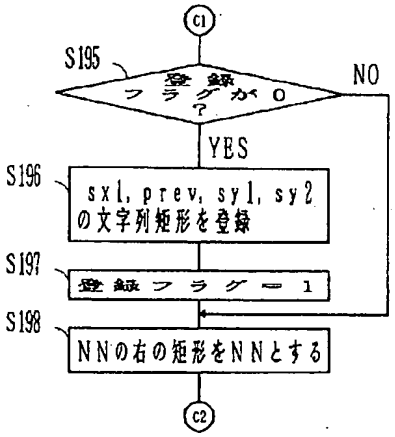
【図６１】

分割された文字列矩形を示す図



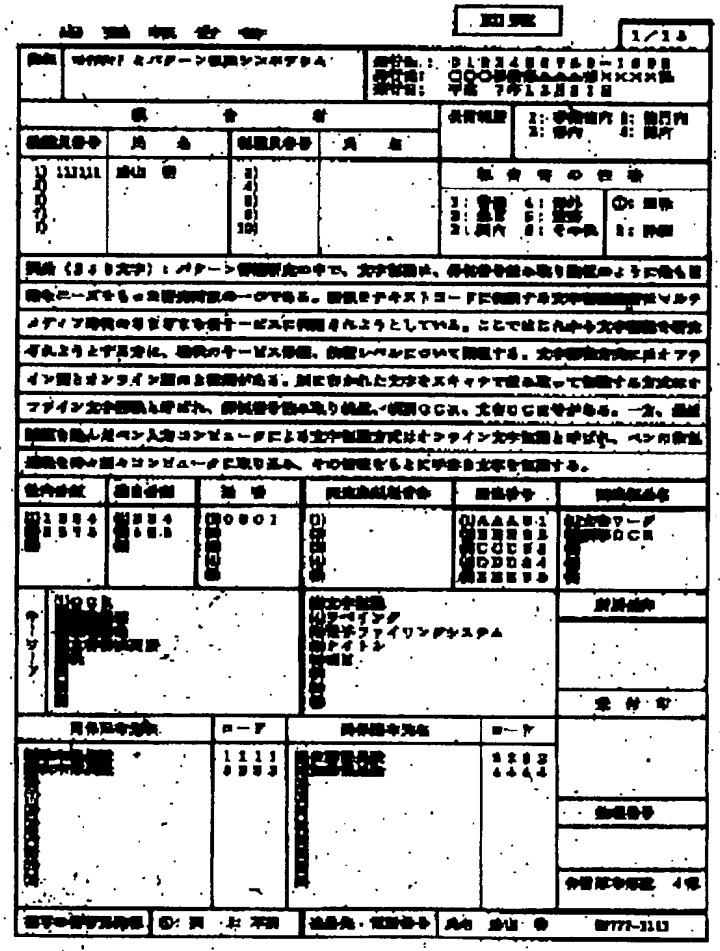
【図６３】

第２の文字列分割処理のフローチャート（その２）



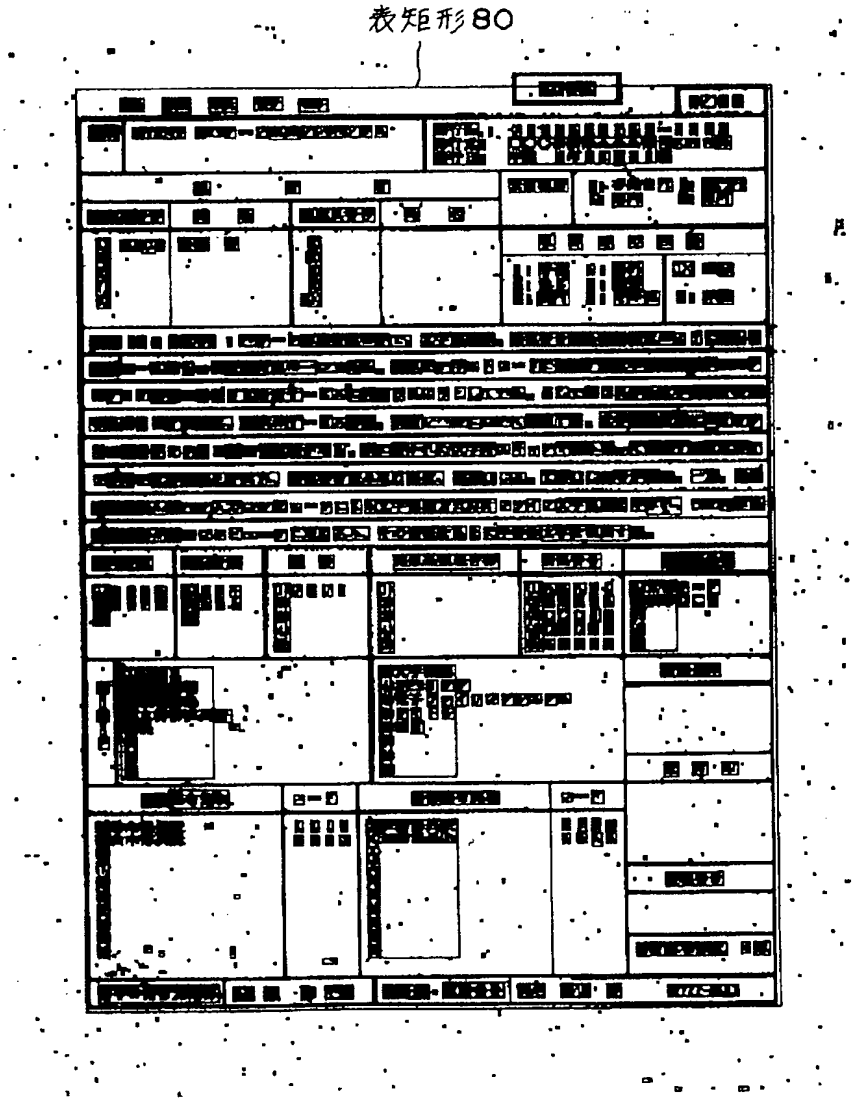
【図５３】

表形式文書の画像データを示す図



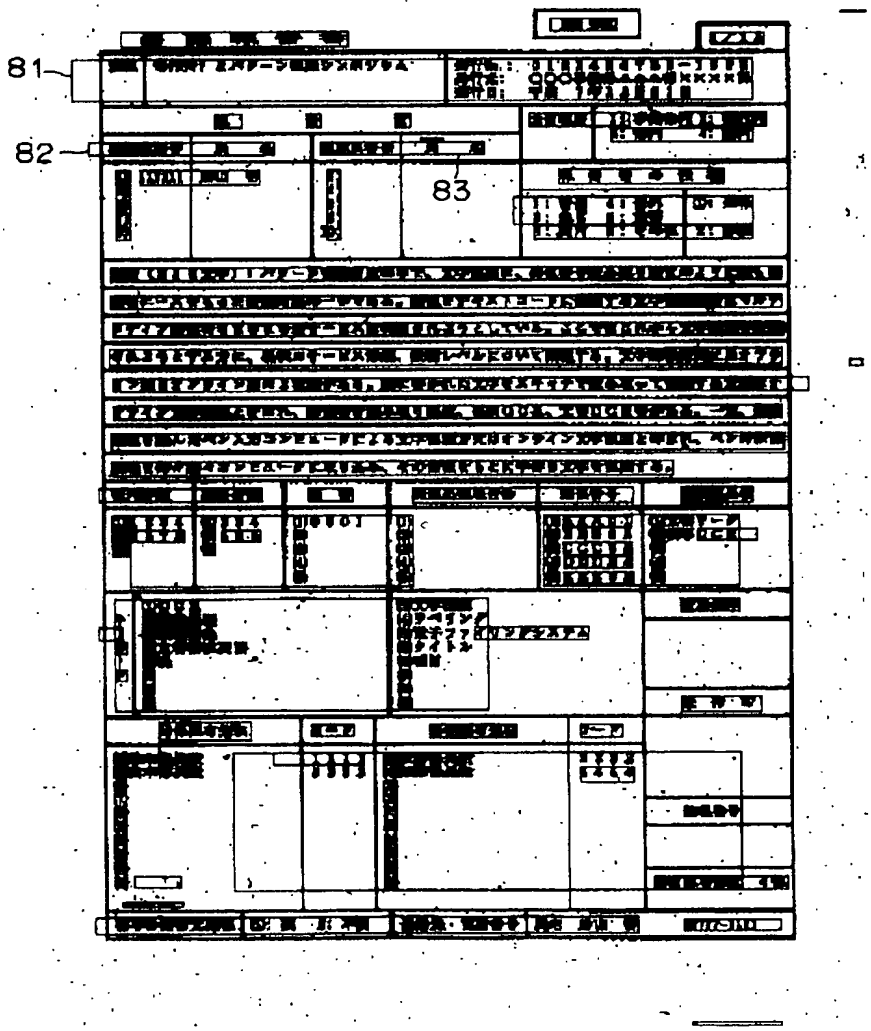
【図５４】

表形式文書のラベリング結果を示す図



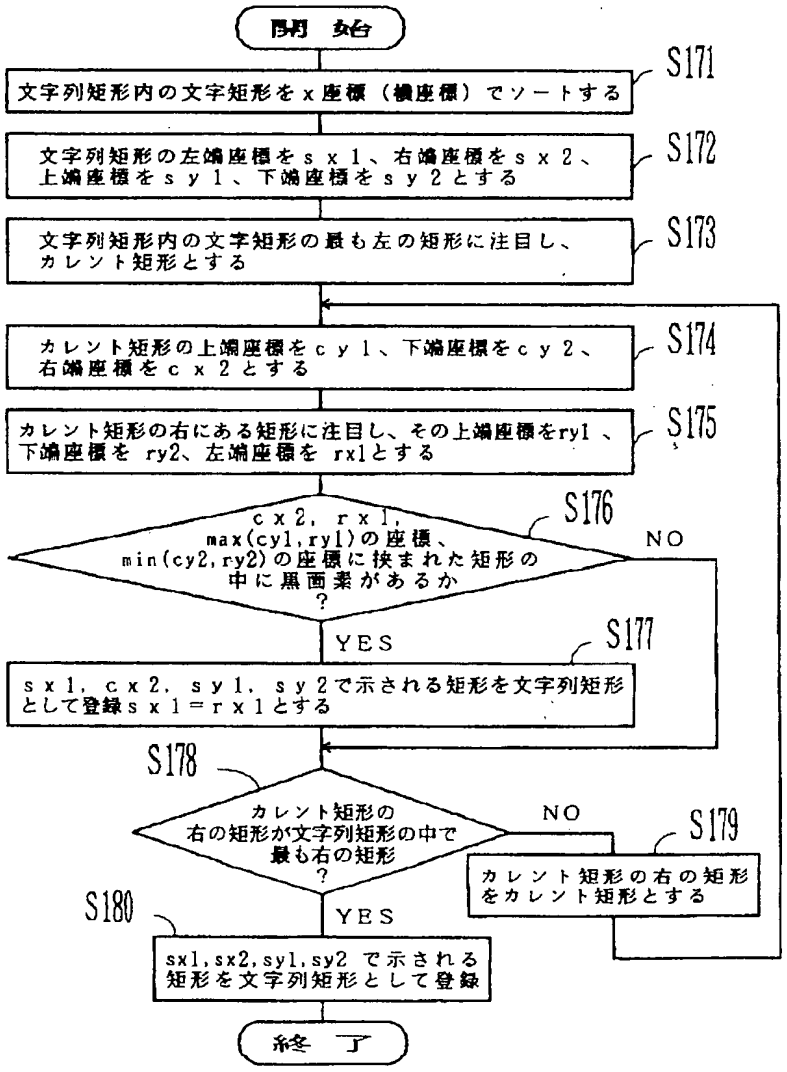
【図５６】

表形式文書の文字列矩形を示す図



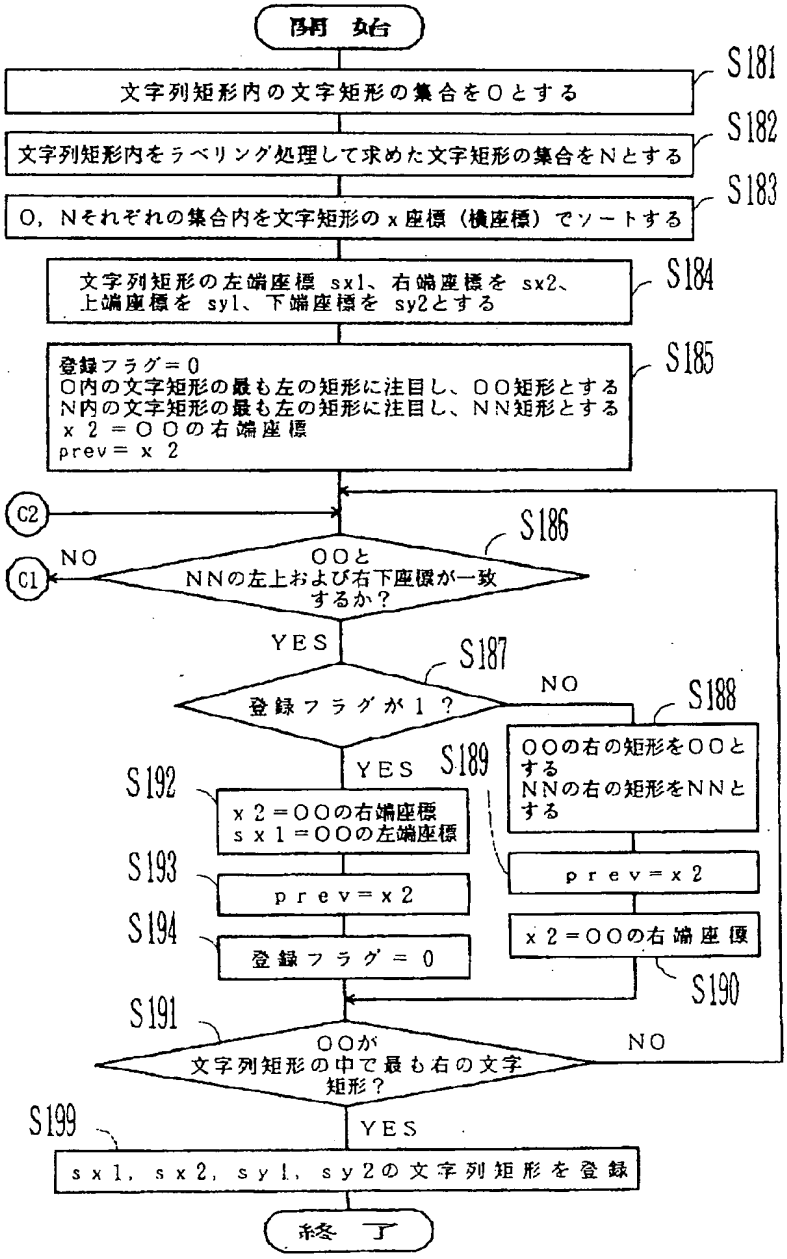
【図５７】

第１の文字列分割処理のフローチャート



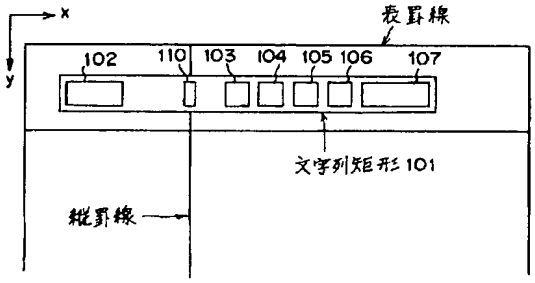
【図６２】

第２の文字列分割処理のフローチャート（その１）



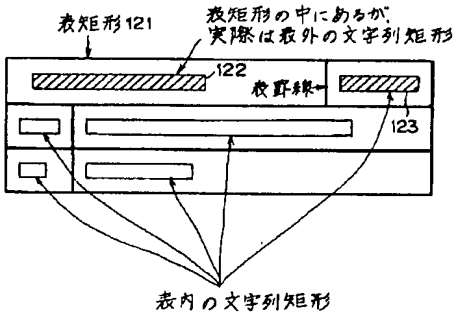
【図６４】

文字列矩形内のラベリング結果を示す図



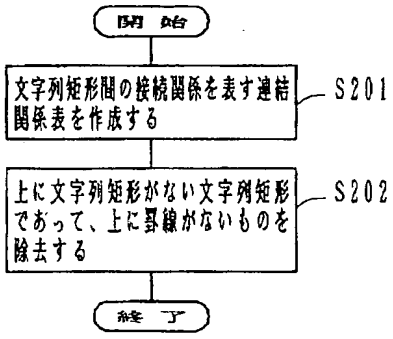
【図６７】

表矩形内の表外文字列矩形を示す図



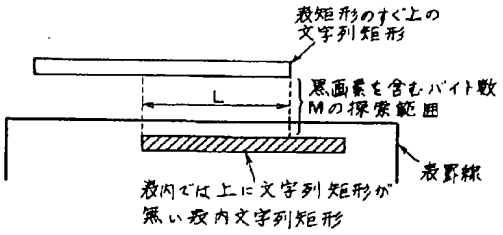
【図６９】

上罫線チェック処理のフローチャート



【図７１】

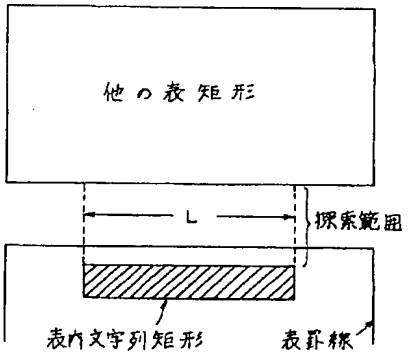
第１の探索範囲を示す図



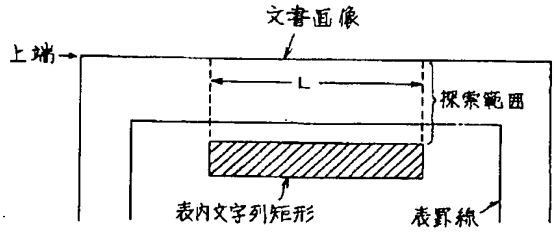
【図７２】

【図７３】

第２の探索範囲を示す図

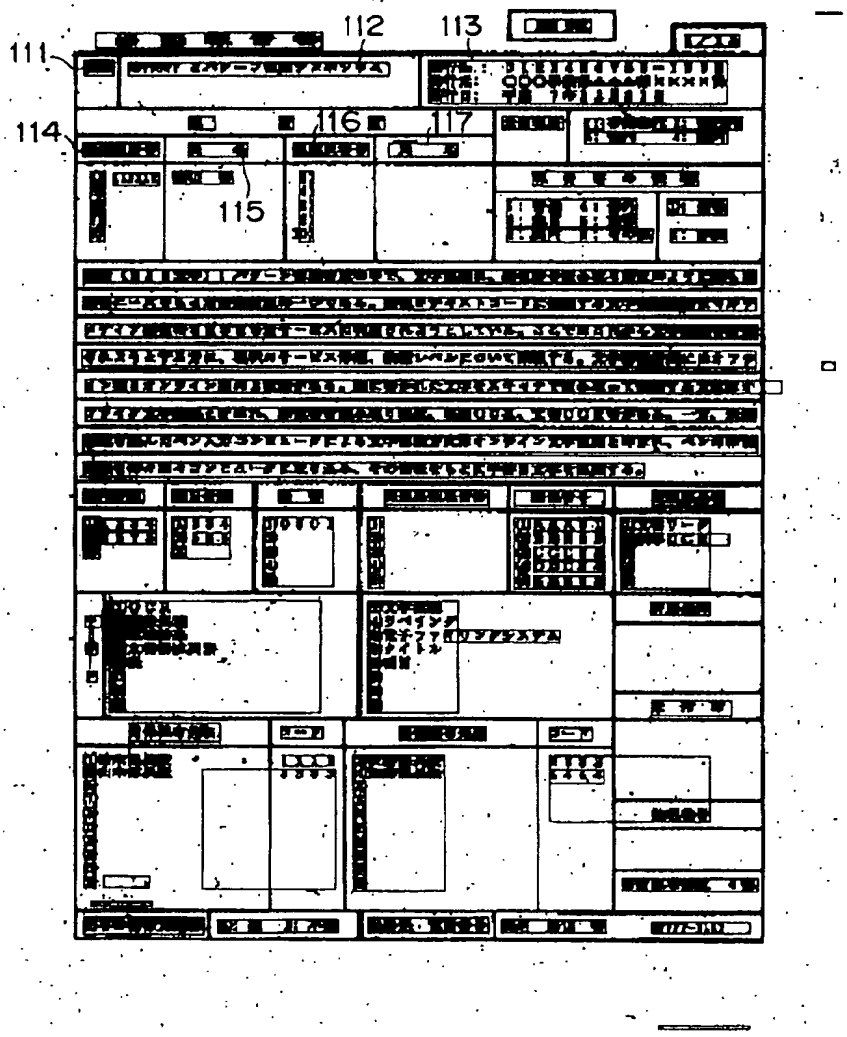


第３の探索範囲を示す図



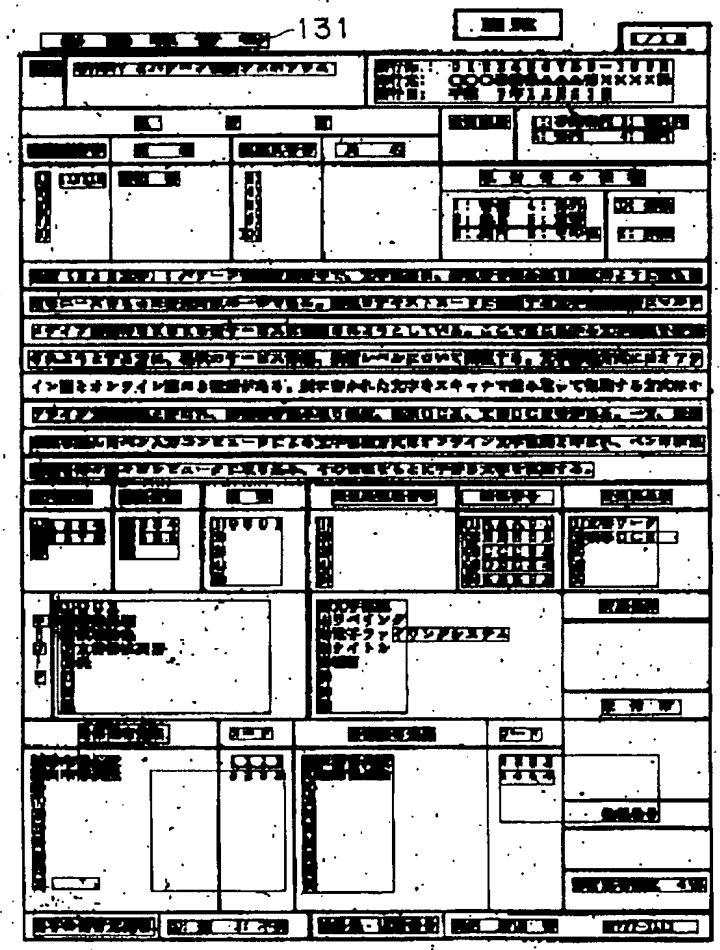
【図６５】

分割処理後の文字列矩形を示す図



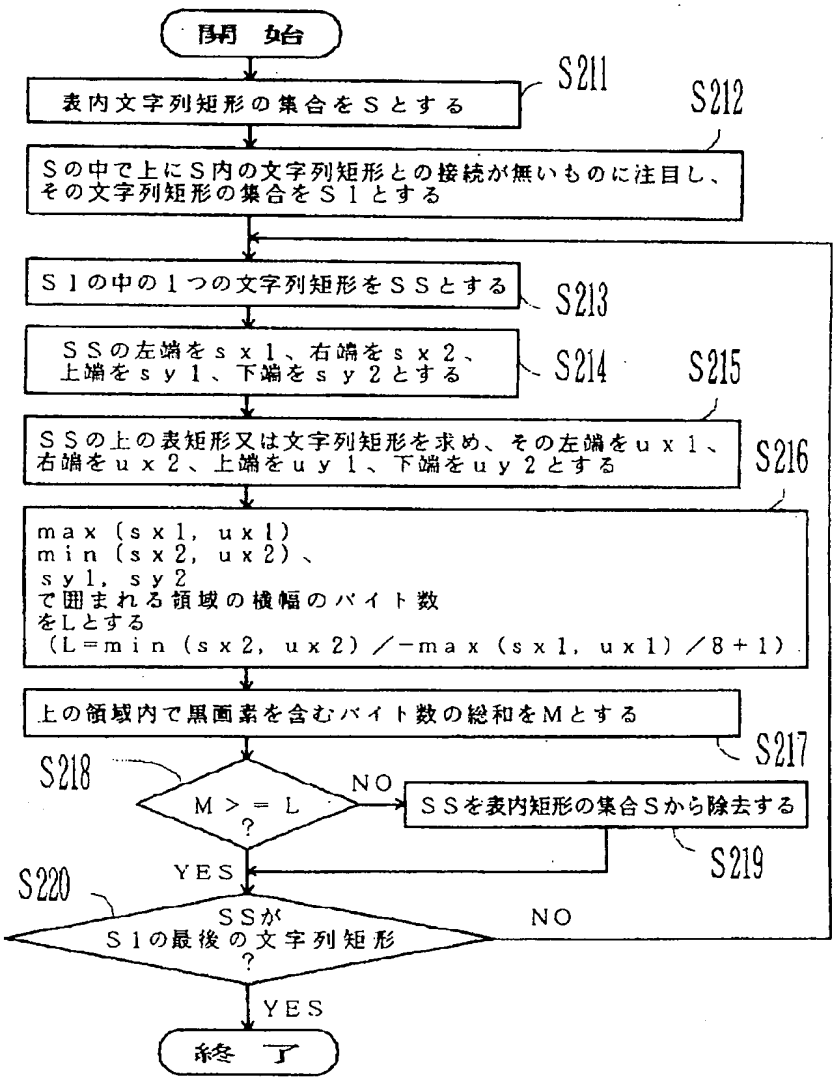
【図６８】

表矩形内の文字列矩形を示す図



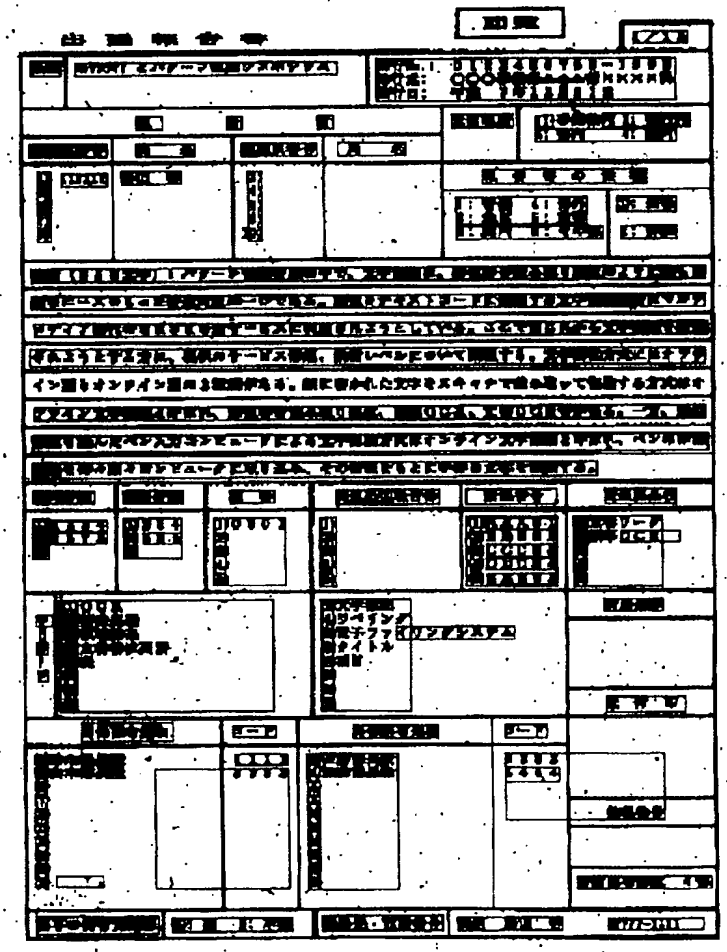
【図７０】

表外文字列矩形除去処理のフローチャート



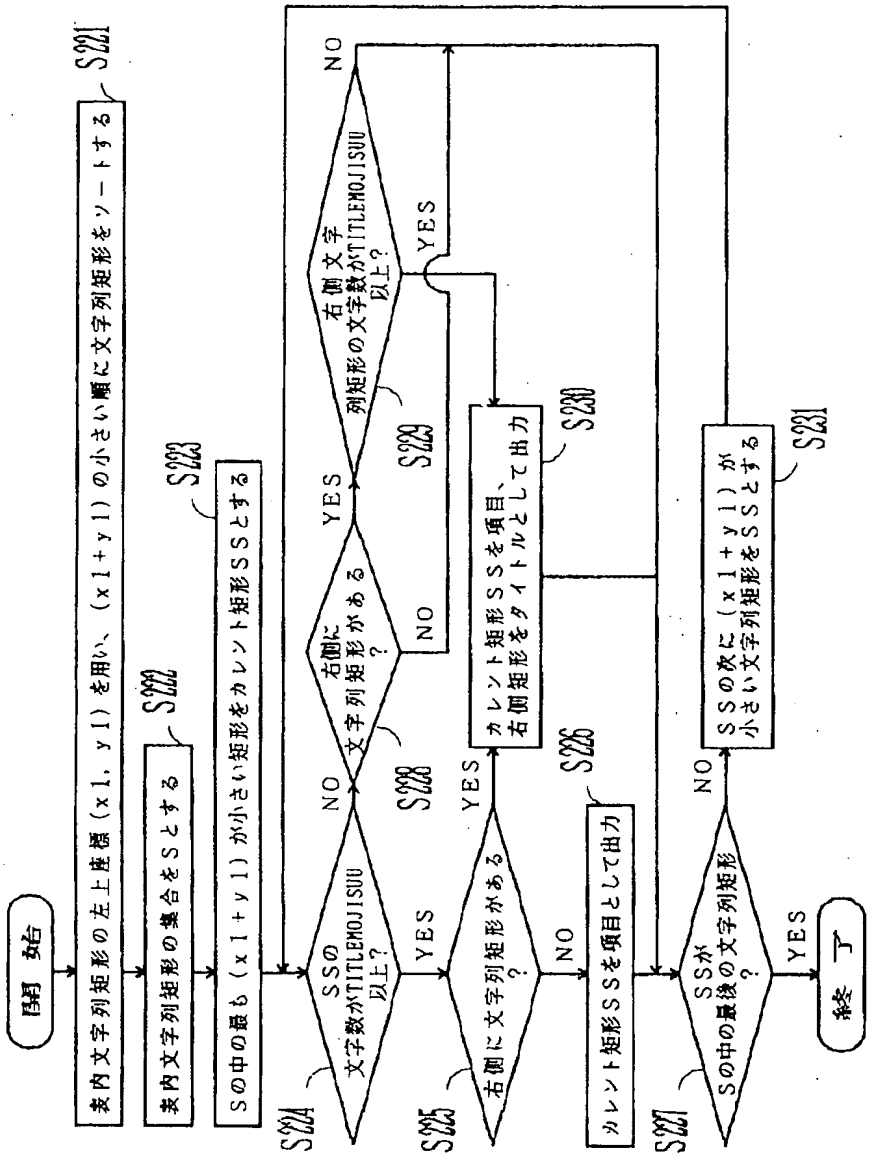
【図７４】

表外文字列矩形除去後の文字列矩形を示す図



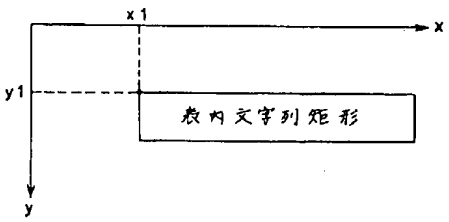
【図７５】

タイトル候補出力処理のフローチャート



【図７６】

文字列矩形の左上頂点の座標を示す図



【図７７】

表内タイトルの抽出結果を示す図